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 | |
| Joshua Ho Gwok Hin | |  | |
| Date issued | Completion date | | Submitted on |
| 11 April 2022 | 14 May 2022 | | 14 May 2022 |
|  | |  | |
| 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: May 2022 |

Content

1. Project background
2. Project Objectives
3. Project Requirement Specifications
4. Task 1 – Software Requirements Specifications
5. Task 2 – Types of Testing
6. Task 3 - Propose relevant tests
7. Task 4 – Project & test scheduling
8. Task 5 – Phase test plan
9. Task 6 – Executing tests

# Project background

“ABC Jobs Pte Ltd” is looking at a Web Application test service for their newly developed community portal.

This jobs community portal are designed for 2 main groups of users. The normal or intended target user group are “Software Programmers” or simply “SP”. This portal is a platform for target user group “SP” interact among themselves.

The second user group are employees of The Company. These “Network Administrators” or “Admin” helps promote the growth of the network as moderators, monitoring “SP” user behaviour and also providing some customer support services.

“Admin” have more access and control over some functions and features of the Web Application.

The Web Application test service covers functional and non-functional tests for different states of the SDLC of this Web Application, where applicable.

# Project Objectives

The main objectives of this testing is to plan do the planning, scheduling and execution of the test for the Web Application.

The outcomes & deliverables are as follows:

* Explain different types of testing used
* Test Schedule Document
* Phase Test Plan
* Execute test scripts and verify results
* Prepare Project Report, Project Presentation and Testing Doc.

# Project Requirement Specifications

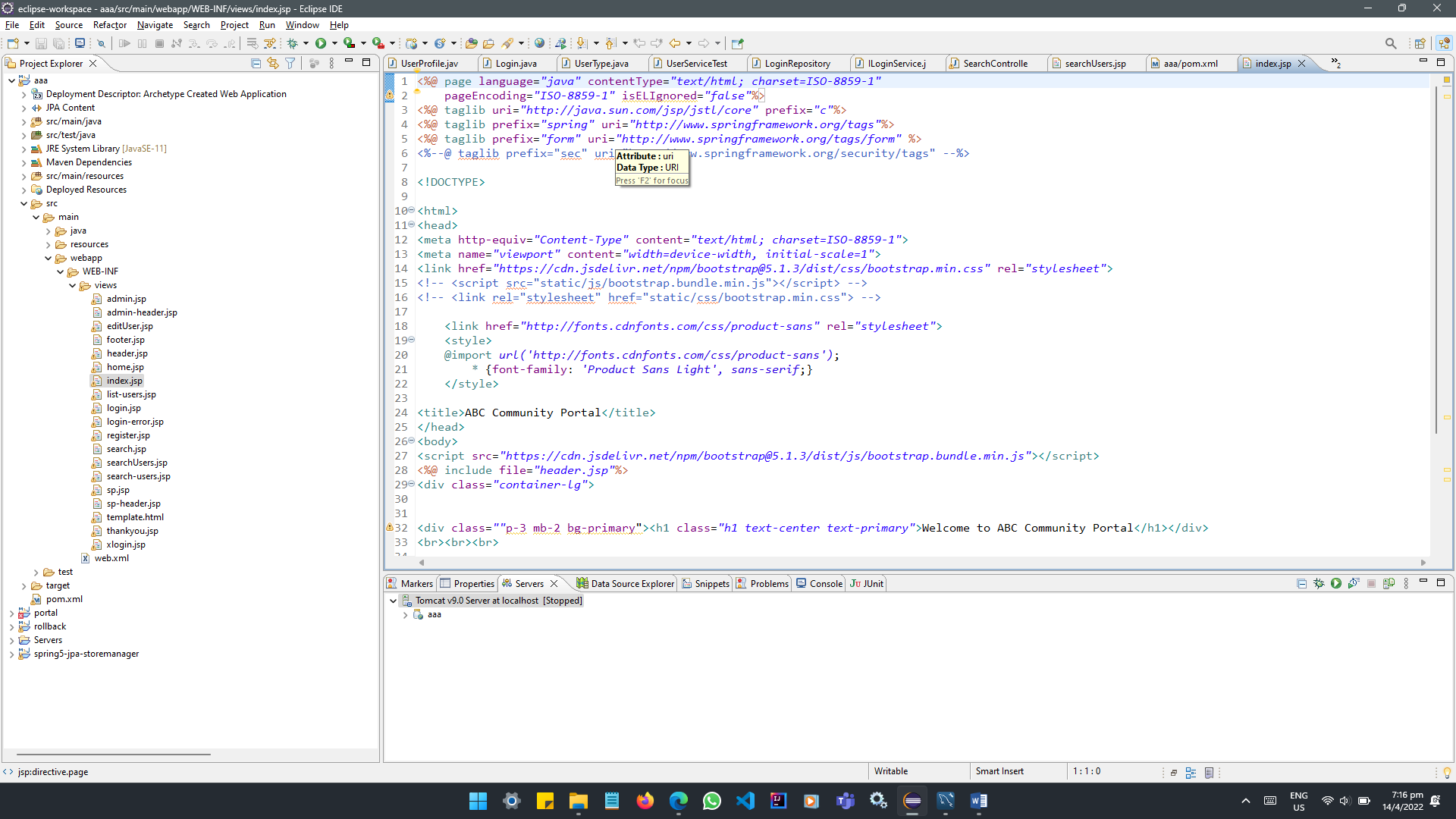
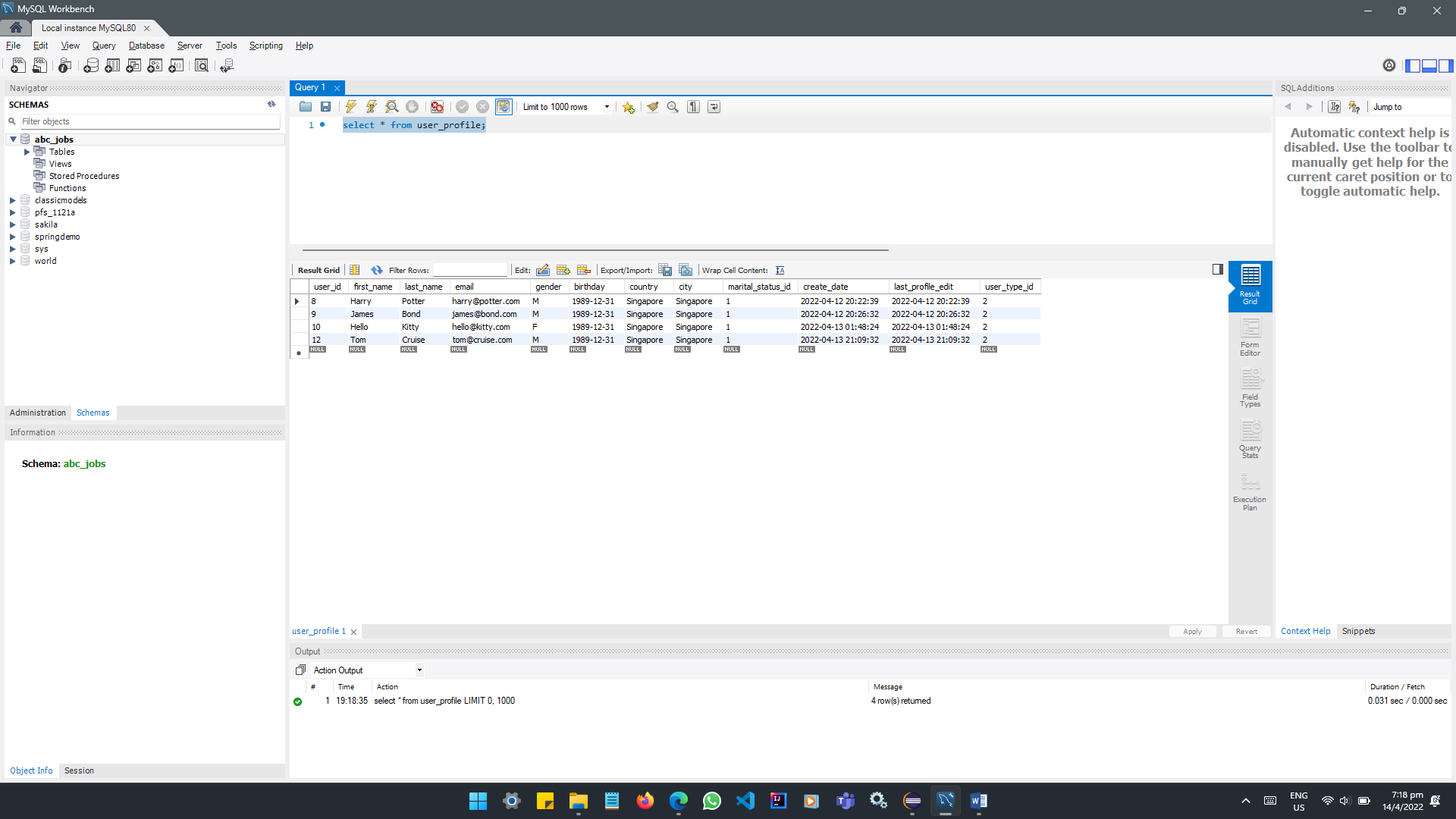
**In this project, we will design, plan and test the community portal website developed in previous module(s). The outcomes & deliverables are as follows:**

* Explain different types of testing used
* Test Schedule Document
* Phase Test Plan
* Execute test scripts and verify results
* Prepare Project Report, Project Presentation and Testing Document

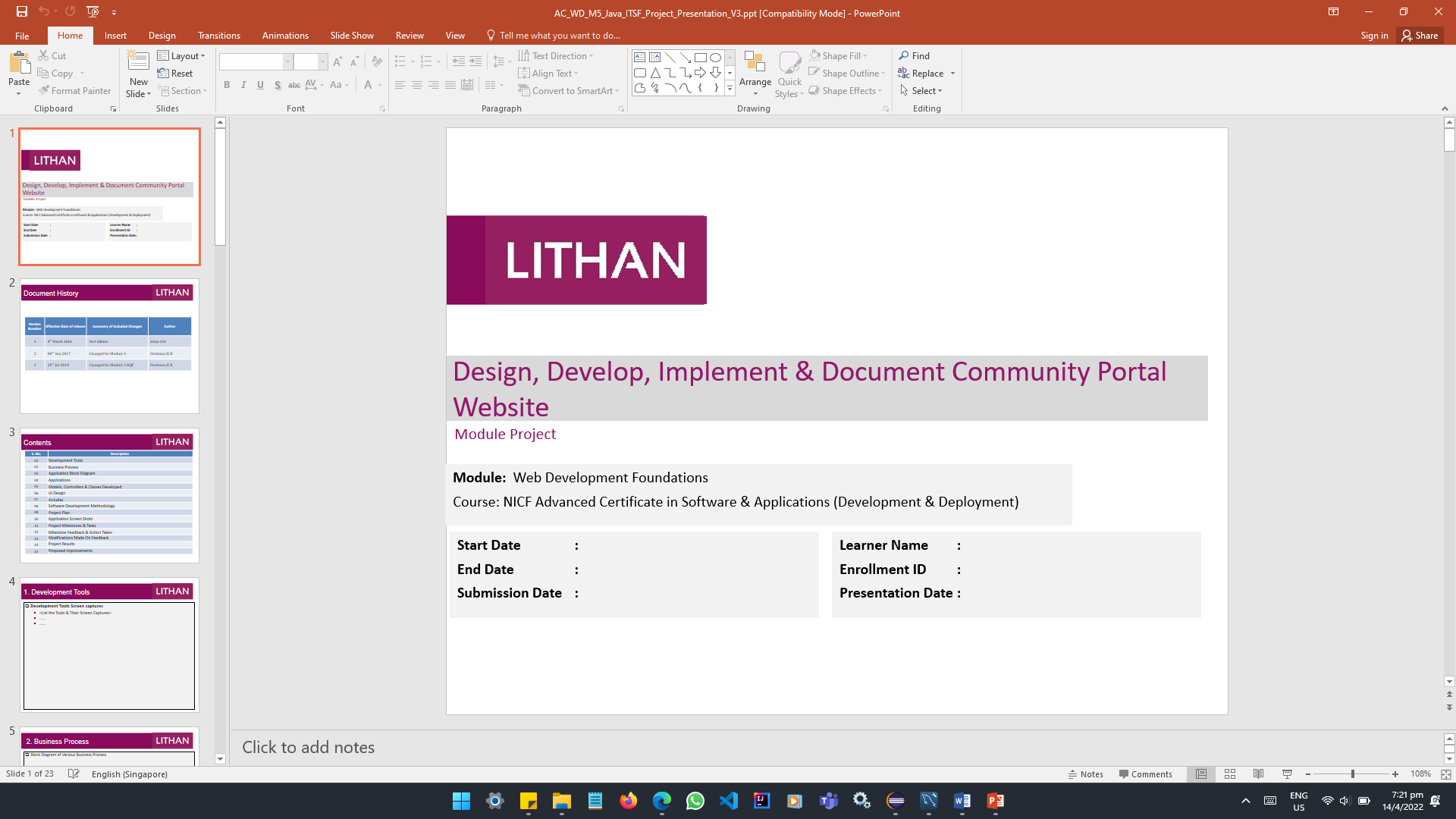
Tools & platform used

* Eclipse IDE - Version: 2021-12 (4.22.0)
* MySQL Workbench 8.0 - Version 8.0.27
* MS Word
* MS PowerPoint
* Tomcat v9.0 Server
* JRE System Library [JavaSE-11]
* Spring - Version 5.1.5.release
* Spring JPA – Version 2.1.5.release
* Hibernate – Version 5.4.1.final
* Junit Jupiter – Version 5.8.2
* Javax Servlet – Version 4.0.1
* JSTL for JSP – Version 1.2.6 (Jakarta)
* ERD - erdplus.com
* JSP, Java, HTML5, JavaScript, CSS3
* Bootstrap – Version 5.1.3

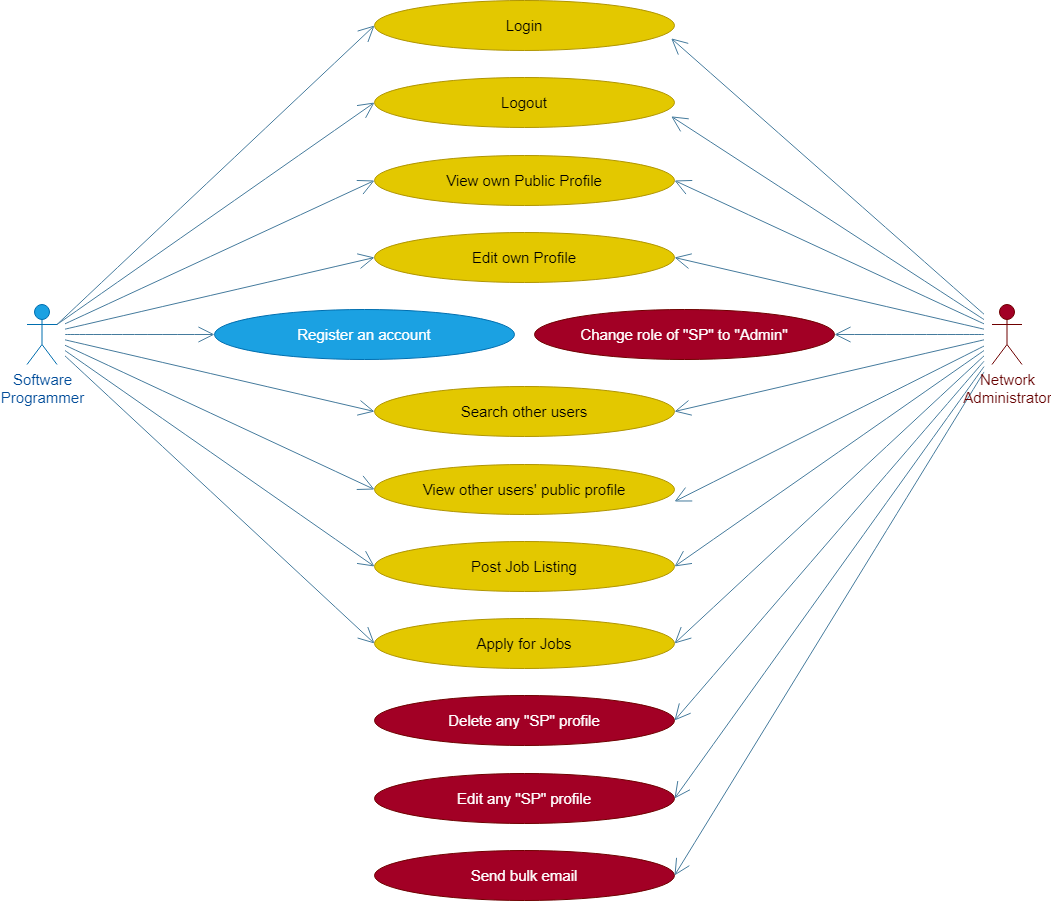
**Eclipse IDE MySQL Workbench 8.0**

**MS Word MS PowerPoint**

**Use case diagram**



**UNIT TESTING**

Each Module/component is tested. Ensure functional correctness and completeness of individual program units. Scope is smaller, easier to fix errors.

More reliable code and less time to find and fix the bugs. Unit tests are easier to debug and only the latest changes need to be debugged if the test fails. We can test a part of the project without waiting for others to be completed due to modular nature of the unit test.

**USABILITY TESTING**

Focuses on the users ease of use of the application, flexibility in handling controls and the ability of the system to meet its objectives. Ensure the user friendliness of the web application.

Helps understand how real users interact with the Web Application and make design changes based on the results. It helps discover undesirable designs even before the actual development of the project.

**CROSS BROWSER TESTING**

Consistency in display and functionality across different browsers.

Ensure design and functionality is working satisfactory across browsers.

**USER ACCEPTANCE TESTING**

Performed at client location by people who are not employees of the organisation that developed the software.

Ensure business process works satisfactorily and met Service-level agreement.

# Task 1 – Software Requirements Specifications

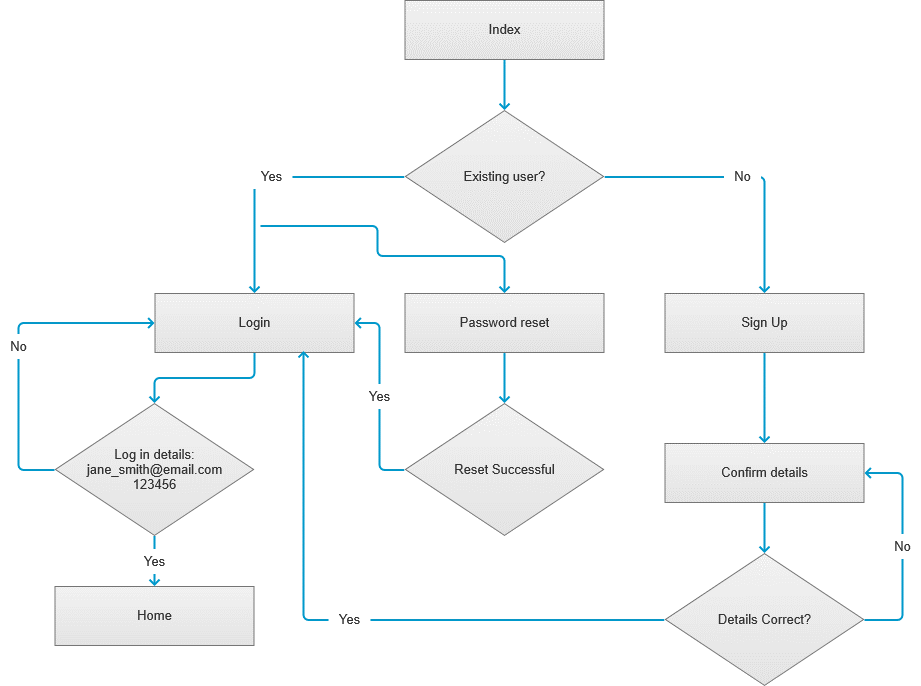
**BUSINESS REQUIREMENTS SPECIFICATIONS**

The Web-Application allows users (SP and Admin) to perform operations of a jobs community portal which includes but not limited to login, logout, register, view profile, edit profile, connect with other users, list or apply for jobs.

Admin have additional access to send bulk email, change an SP to Admin, edit and delete other SP.

|  |  |
| --- | --- |
| **Use cases** | **Non-Functional Business Requirements** |
| **Software Programmers (SP):**  SP – Login  SP – Logout  SP – Register  SP – View own Public Profile  SP – Edit details of own Profile  SP – Search other users  SP – View other user’ Public Profile  SP – Post Job Listing(s)  SP – Apply for Job(s)  **Network Administrators (Admin):**  Admin – Login  Admin – Logout  Admin – View own Public Profile  Admin – Edit details of own Profile  Admin – Search other users  Admin – View other user’ Public Profile  Admin – Post Job Listing(s)  Admin – Apply for Job(s)  Admin – Edit, Delete other SP  Admin – Allow SP to have Admin rights  Admin – Send Bulk email | Loading time of all pages within 5 seconds  Request & Response of many concurrent users. |

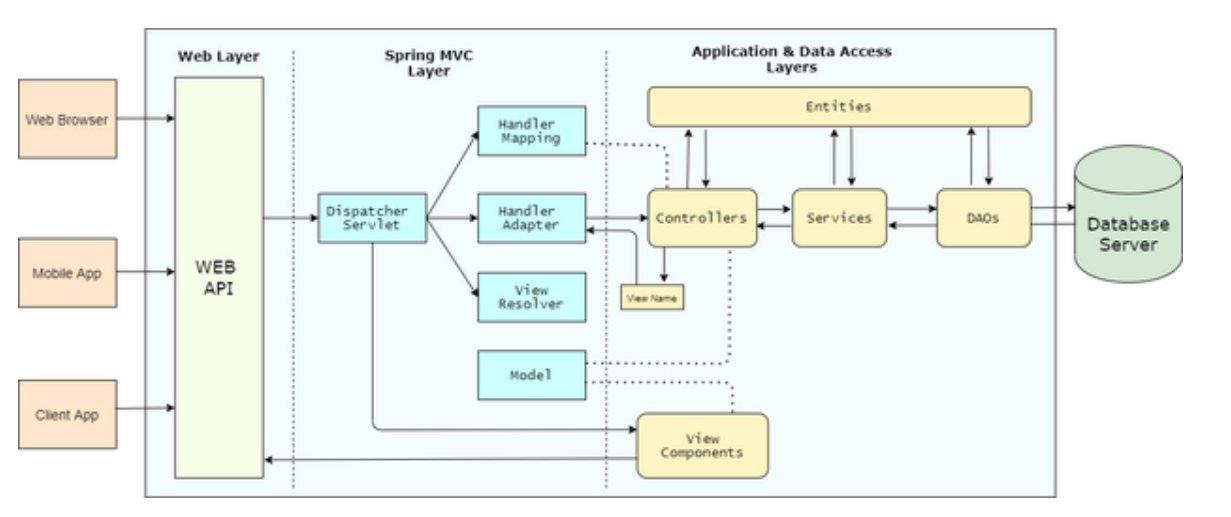
Business Process Diagram (User access)

****

**ARCHITECTURAL & DESIGN REQUIREMENTS**

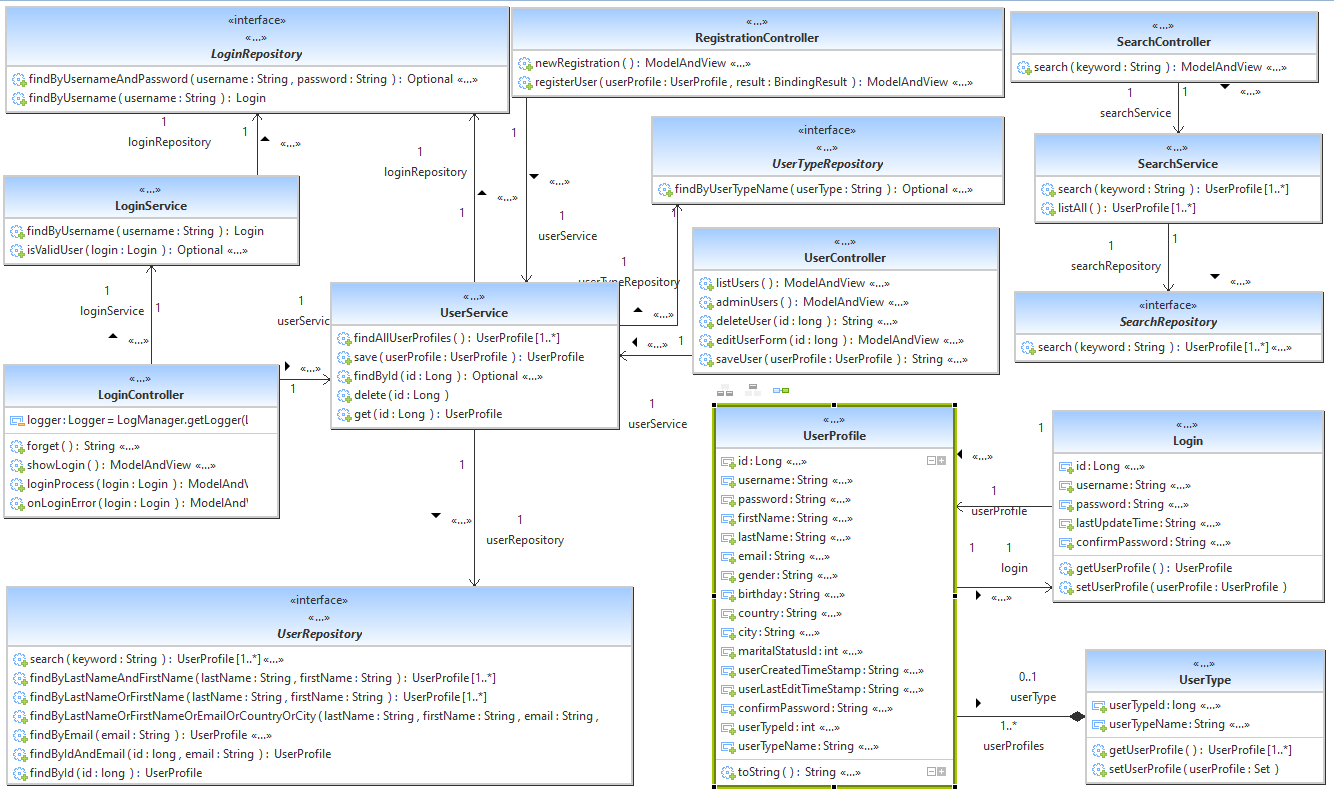
MVC Design Pattern -

Built based on Spring Framework’s Model, Views and Controller model.



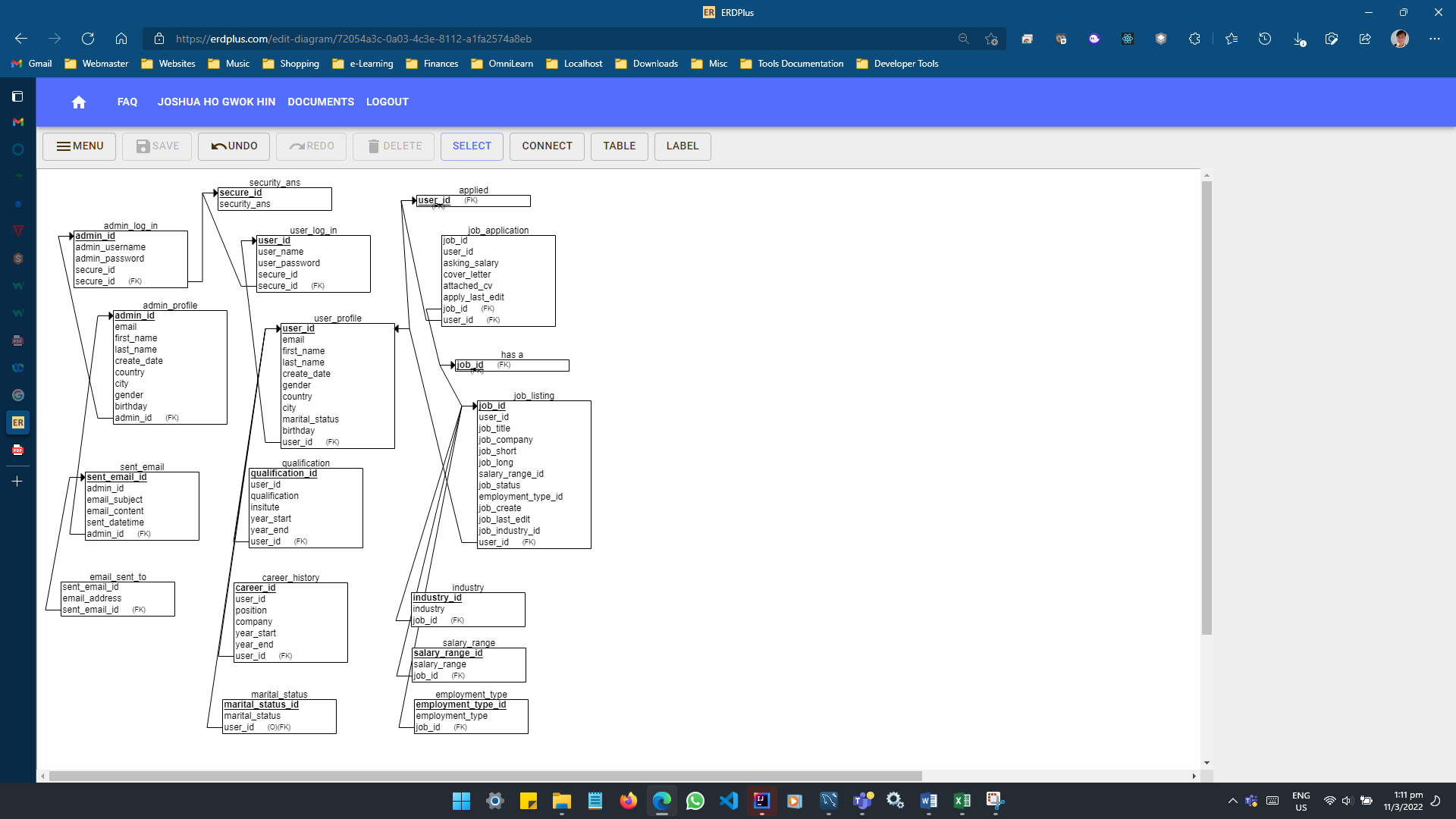
Class Diagram –

Shows the Object Relational Mapping of classes

****

Database Entity Relationship Diagram –

Database tables and columns



**SYSTEM & INTEGRATION REQUIREMENTS**

|  |
| --- |
| **DEVELOPMENT & STAGING ENVIRONMENT**  Operating Systems   * Windows 11 Home - version 22H2, OS build 22610.1 * System Type 64-bit Operating System   Hardware   * Processor: Intel(R) Core(TM) i5-10300H * RAM: 16 GB * x64-based processor   Development Tools   * Java Development Kit (JDK) 11.0.3 * Java(TM) SE Runtime Environment (build 1.8.0\_331-b09) * Apache Tomcat v9.0 * Eclipse IDE for Enterprise Java and Web Developers – Version 2021-12(4.22.0) * MySQL Workbench - build 8.0.27 * Web browsers with JavaScript enabled * Firefox 98, Edge 99 & Chrome 99   Test to be conducted in this environment:   * Unit testing * Cross browser testing * Usability Testing |
| **TARGET PRODUCTION ENVIRONMENT**  Operating Systems   * Windows 10 or later * Mac OS Big Sur (11) or later * Android 11 or later * IOS 11 or later   Software   * Web browsers with JavaScript enabled * Chrome 93 or later, Edge 16 or later, Firefox 88 or later   Hardware   * Processor: x86 or x64 * RAM: 1 GB (min), 4 GB (recommended)   Connection   * Internet connectivity of ≥ 1 mbps download and upload   Test to be conducted in this environment:   * User Acceptance Testing |

Chart

Description automatically generated

# Task 2 – Types of testing

|  |  |  |
| --- | --- | --- |
| **Types of Testing** | **Functional/ Non-functional** | **Brief description & it’s utility** |
| Unit testing | Functional | Each Module/component is tested. Ensure functional correctness and completeness of individual program units. Scope is smaller, easier to fix errors. |
| Integration testing | Functional | Testing interconnectivity between the different components in a system. Helps expose faults in the interaction between integrated units. Required test environment to be set up for the test. Ensure all high prioritized bugs are fixed and closed. |
| System testing | Functional | Testing of a complete and fully integrated software product. Enables to test, verify and validate the business requirements and applications. Application is tested to verify the technical and functional specifications. |
| System testing  Usability testing | Non-functional | Focuses on users ease of use of the application, flexibility in handling controls and that the ability of the system to meet its objectives. |
| System testing  Regression testing | Functional | Retesting previously tested components to ensure proper functionality after modification in a part of system. |
| System testing  Recovery testing | Non-functional | Demonstrates a software solution is reliable and trustful. |
| System testing  Stress testing | Non-functional | Used to test the stability & reliability of the system. Stress tests helps determine the limit of where a system software or hardware breaks, the system robustness and error handling in heavy load condition. Provides effective error management under extreme conditions. |
| System testing  Load testing | Non-functional | Determines a systems performance under real-life load conditions. Provides application behaviour when multiple users access it simultaneously. Load testing helps identifies max operating capacity, current infrastructure sufficiency, sustainability of app with respect to peak user load, number of users an app can support and how scalable to allow more users to access. |
| Interface testing | Functional | Interfaces between system components are tested. Ensure absence of incorrect mapping of data. Interfaces are tested in two phases. Individually during system testing and tested together during integration testing. Can help identify user-friendliness and verify security requirements while communication propagates between the systems. |
| Cross browser test | Non -functional | Consistency in display and functionality across different browsers. |
| Acceptance testing | Functional | Validate the end to end business flow that no critical defects and business process works satisfactorily. |
| Acceptance testing  Internal (Alpha) | Functional | Performed by members of the organisation that developed the software but who are not directly involved in the project. |
| Acceptance testing  External (Beta)/UAT | Functional | Performed at client location by people who are not employees of the organisation that developed the software. |

# Task 3 – Propose relevant tests

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Ref** | **Function/Page** | **Functional Test** | **Non-functional Test** |
| UC01 | Login | Unit Testing, UAT | Usability testing, Cross browser testing |
| UC02 | Register | Unit Testing, UAT | Usability testing, Cross browser testing |
| UC03 | View own Profile | Unit Testing, UAT | Usability testing, Cross browser testing |
| UC04 | Update own Profile | Unit Testing, UAT | Usability testing, Cross browser testing |
| UC05 | Search other users’ profile | Unit Testing, UAT | Usability testing, Cross browser testing |
| UC06 | View other user’s Public Profile | Unit Testing, UAT | Usability testing, Cross browser testing |
| UC07 | Password reset |  | Usability testing |

Usability testing (Non-functional)

Focuses in the users ease of use of the application, flexibility in handling controls and the ability of the system to meet its objectives.

Helps understand how real users interact with the Web Application and make design changes based on the results. It helps discover undesirable designs even before the actual development of the project.

|  |  |  |
| --- | --- | --- |
| **USABILITY TESTING** | | |
| Use Case Ref | Use Case | Benefits |
| UST01 | Login | Login design is understood and perform with ease |
| UST02 | Register | Logical layout of fields, user are not confused |
| UST03 | View own Profile | User able to figure out how to view own profile |
| UST04 | Update own Profile | Ensure update of profile task can be accomplished easily |
| UST05 | Search other users’ profile | Design is intuitive and simple for search function |
| UST06 | View other user’s Public Profile | User can instinctively view a profile after search results. |
| UST07 | Password reset | Flow of events are logical without confusion |

Unit testing (Functional)

Each Module/component is tested. Ensure functional correctness and completeness of individual program units. Scope is smaller, easier to fix errors.

More reliable code and less time to find and fix the bugs. Unit tests are easier to debug and only the latest changes need to be debugged if the test fails.

|  |  |  |
| --- | --- | --- |
| **UNIT TESTING** | | |
| Use Case Ref | Use Case | Benefits |
| UNT01 | Login | Scope is smaller, easier to fix errors |
| UNT02 | Register | Scope is smaller, easier to fix errors |
| UNT03 | View own Profile | Scope is smaller, easier to fix errors |
| UNT04 | Update own Profile | Scope is smaller, easier to fix errors |

Cross browser testing (Non-functional)

Consistency in display and functionality across different browsers.

Ensure design and functionality is working satisfactory across browsers.

|  |  |  |
| --- | --- | --- |
| **CROSS BROWSER TESTING** | | |
| Test Case ID | Pages | Benefits |
| CBT01 | index.jsp | Layout and functions work across browsers |
| CBT02 | login.jsp | Layout and functions work across browsers |
| CBT03 | home.jsp | Layout and functions work across browsers |
| CBT04 | searchUsers.jsp | Layout and functions work across browsers |

User acceptance testing (Functional)

Performed at client location by people who are not employees of the organisation that developed the software.

Ensure business process works satisfactorily and met Service-level agreement.

|  |  |  |
| --- | --- | --- |
| **USER ACCEPTANCE TESTING** | | |
| Use Case Ref | Use Case | Benefits |
| UAT01 | Software Engineers | Users are able to verify that login feature is working |
| UAT02 | Network Administrators | Users are able to test that register feature is working |

# Task 4 – Project & test scheduling

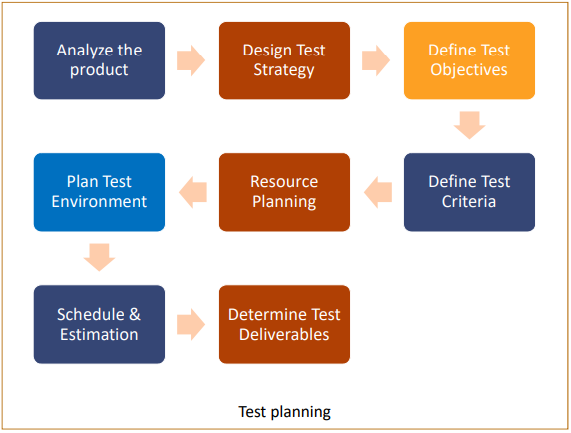
|  |  |  |  |
| --- | --- | --- | --- |
| **TEST SCHEDULE** | | | |
| Test Schedule ID | 01 | | |
| Product Name | ABC Jobster Community Portal | | |
| Product Version | Jobster 1.0 | | |
| Present owner | Joshua Ho Gwok Hin | | |
| Created on | 09 May 2022 | | |
| Review on | 13 May 2022 | | |
| Review by | Nas | | |
| Review comments |  | | |
| Current version | Jobster TP 1.0 | | |
| Change details |  | | |
| Current status | Draft/ In Process/ Approved | | |
| Signing Off Authority | Name | Position | Signature &  Date |
|  |  | Team Manager |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Activity** | **Start** | **End** | **Person days** | **Resource** |
| Prepare Test Plan | 1 May 2022 | 1 May 2022 | 1 | Development Team |
| Review Test Plan | 2 May 2022 | 2 May 2022 | 1 | Development Team |
| Prepare Test Scripts | 3 May 2022 | 3 May 2022 | 1 | Development Team |
| Setup Test environment | 4 May 2022 | 4 May 2022 | 1 | Development Team |
| Maintain test environment | 5 May 2022 | 5 May 2022 | 1 | Development Team |
| Establish Test Procedures | 6 May 2022 | 6 May 2022 | 1 | Development Team |
| Execute Testing/ Record faults | 7 May 2022 | 7 May 2022 | 1 | Development Team |
| Certify App to be released to Beta Test Group | 8 May 2022 | 8 May 2022 | 1 | Development Team Lead |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Features** | **Weighting** | **Functional Points** | **Total Weighting** | **Input** | **Process** | **Output** |
| Login | 1 | 6 | 6 | username, password,  loginProcess() | findByUsername(), isValidUser() | home.jsp |
| Logout | 1 | 3 | 3 | logout() | invalidateHttpSession() | showLogin.jsp |
| Register | 5 | 13 | 65 | username, password,  confirm password, first name,  last name,  email,  gender,  birthday  marital status, country,  city  register() | registerUser() | login.jsp |
| View Public Profile | 1 | 4 | 4 | viewPublicProfile() | findById(),  viewUserProfile() | public-profile.jsp |
| Search Profile | 2 | 4 | 8 | keyword,  search() | List<UserProfile> | searchUsers.jsp |
| Edit Profile | 5 | 14 | 70 | username, password,  confirm password, first name,  last name,  email,  gender,  birthday  marital status, country,  city  updateProfile() | save() | update-success.jsp |
| Reset Password | 5 | 7 | 35 | email,  findByEmail() | isValidUser(),  resetPasswordEmail()  validateReset(),  saveNewPassword() | showLogin.jsp |
|  | | | | | | |
| Total Functional Points | | | | | | 191 |
| Hours per Functional Points | | | | | | 0.5 |
| Total Person hours | | | | | | 95.5 |
| Number of days | | | | | | 12 |

# Task 5 – Phase test plan

**Test planning process**

****

**Analyse the product**

The first step towards creating a test plan is to analyse the product, its features and functionalities to gain a deeper understanding. Further, explore the business requirements and what the client wants to achieve from the end product. Understand the users and use cases to develop the ability of testing the product from user’s point of view.

**Design the Test Strategy**

Once you have analysed the product, you are ready to develop the test strategy for different test levels. Your test strategy can be composed of several testing techniques. Keeping the use cases and business requirements in mind, you decide which testing techniques will be used.

**Define the Test Objectives**

A good test plan clearly defines the testing scope and its boundaries. You can use requirements specifications document to identify what is included in the scope and what is excluded. Make a list of ‘Features to be tested’ and ‘Features not to be tested’. This will make your test plan specific and useful

**Define Test Criteria**

Test Criteria is a standard or rule on which a test procedure or test judgment can be based. There’re 2 types of test criteria as following:

* Suspension Criteria
* Exit Criteria

**Resource Planning**

* The resource plan is a detailed summary of all types of resources required to complete the project task. The resource could be human, equipment and materials needed to complete a project
* The resource planning is an important to factor of the test planning because helps in determining the number of resources (employee, equipment…) to be used for the project.

**Plan Test Environment**

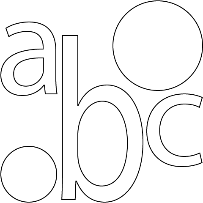
A testing environment is a setup of software and hardware on which the testing team is going to execute test cases

**Schedule & Estimation**

In the Test Estimation phase, the whole project breaks out into small tasks and the estimation for each task has been adding.

**Determine Test Deliverables**

Test Deliverables is a list of all the documents, tools and other components that has to be developed and maintained in support of the testing effort.

 Test Plan

Web Application

ABC Jobster Community Portal

**Test Plan Identifier**

Test Plan Serial/ID: TP\_WEB\_APP\_001

Author: Joshua Ho

Testing Period: 2H 2022

Revision History: N.A

**Goal**

The goal of this test is to identify and measure the problems/ issues in the Web Application.

**Objectives**

Validate and verify system functionality works according to the requirements.

The test activities will be performed by based on the Software Requirements Specifications contained in this document:

1. [BUSINESS REQUIREMENTS SPECIFICATIONS Page 6](#BRS)
2. [ARCHITECTURAL & DESIGN REQUIREMENTS Page 7](#Design)
3. [SYSTEM & INTEGRATION REQUIREMENTS Page 9](#Sys)

|  |  |  |
| --- | --- | --- |
| **Use Case** | **Functional Test** | **Non-functional Test** |
| Login | Unit Testing, UAT | Usability testing, Cross browser testing |
| Register | Unit Testing, UAT | Usability testing, Cross browser testing |
| View own Profile | Unit Testing, UAT | Usability testing, Cross browser testing |
| Update own Profile | Unit Testing, UAT | Usability testing, Cross browser testing |
| Search other users’ profile | Unit Testing, UAT | Usability testing, Cross browser testing |
| View other user’s Public Profile | Unit Testing, UAT | Usability testing, Cross browser testing |
| Password reset |  | Usability testing |

**Features to be tested**

1. Login
2. Logout
3. Registration
4. View Public Profile
5. Search Profile
6. Update Profile
7. Reset password

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST CASES** | | | | |
| No. | Test Type | Test Case ID | Test Item |
| 01 | Usability Testing | UST01 | Register |
| 02 | Usability Testing | UST02 | Login |
| 03 | Usability Testing | UST03 | View Profile |
| 04 | Usability Testing | UST04 | Update Profile |
| 05 | Usability Testing | UST05 | Search other users |
| 06 | Usability Testing | UST06 | View other user’ Public Profile |
| 07 | Usability Testing | UST07 | Log out |
| 08 | Unit Testing | UNT01 | Login |
| 09 | Unit Testing | UNT02 | View Profile |
| 10 | Unit Testing | UNT03 | Update Profile |
| 11 | Unit Testing | UNT04 | Search other users |
| 12 | Cross Browser Testing | CBT01 | index.jsp |
| 13 | Cross Browser Testing | CBT02 | login.jsp |
| 14 | Cross Browser Testing | CBT03 | home.jsp |
| 15 | Cross Browser Testing | CBT04 | searchUsers.jsp |
| 16 | User Acceptance Testing | UAT01 | Software Engineers |
| 17 | User Acceptance Testing | UAT02 | Network Administrators |

**Features not to be tested**

1. Post Job
2. Apply for Job
3. Change role of “SP” to “Admin”
4. Delete any “SP” profile
5. Edit any “SP” profile
6. Send bulk email

**Approach**

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

**Suspension Criteria**

1. Build containing serious defects which limit testing progress.
2. Significant change in requirements suggested by client.
3. Software/hardware problems.
4. Assigned resources are not available when needed.

**Resumption Criteria**

1. All issues in suspension criteria have been resolved or mitigated
2. New software build has been redeployed
3. New build with fixed critical

**Exit Criteria**

1. All High and Medium priority requirements have been tested without Critical or Medium severity defects.

**Test Metrics**

|  |  |  |
| --- | --- | --- |
| TEST METRIC |  | TESTING PHASE |
| Number of test cases | 17 | Test Development Phase |
| Number of Test cases Passed | 17 | Test Execution Phase |
| Number of Test cases Executed | 17 | Test Execution Phase |
| Number of Test cases Failed | 0 | Test Execution Phase |
| Number of Test cases under Investigation | 0 | Test Development Phase |
| Number of Test cases Blocked | 0 | Test Development Phase / Execution Phase |
| Number of Test cases Re- executed | 0 | Regression Phase |
| Number of First run Failures | 0 | Test Execution Phase |
| Total Executions | 17 | Test Reporting Phase |
| Total Passes and Failures | 17 | Test Reporting Phase |
| Test case Execution time | 30min | Test Reporting Phase |
| Test Execution time | 30min | Test Reporting Phase |

**Test Deliverables**

1. Phase Test Plan
2. Test Schedules
3. Test Scripts
4. Test Results
5. Bugs Write-up

# Task 6 – Executing tests

**Usability Testing**

To begin test, visit [https://pj3jn7.axshare.com/#id=h1uc2r&p=index](https://pj3jn7.axshare.com/%23id=h1uc2r&p=index)

Metrics used:

Time-on-task (TOT): Measured in seconds.

User Error Rate (UER): In percentage. (No of Error/ Total Participants X 100%)

Task Success Rate (TSR): In percentage. (No. of successful/ Total of Participants X 100%)

Net Promoter Score (NPS): A scale of 0 to 10. 0-6 Detractors, 7-8 Passives and 9-10 as Promoters.

Customer Satisfaction (CSAT): 1-2 Very Bad, 3-4 Poor, 5-6 Average, 7-8 Good, 9-10 Excellent

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TC ID** | **Scenarios** | **No of user(s)** | **No of test conducted** | **Facilitator** |
| UST01 | Register | 5 | 5 | Joshua |
| UST02 | Login | 5 | 5 | Joshua |
| UST03 | View Profile | 5 | 5 | Joshua |
| UST04 | Update Profile | 5 | 5 | Joshua |
| UST05 | Search other users | 5 | 5 | Joshua |
| UST06 | View other user’ Public Profile | 5 | 5 | Joshua |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Participants** | **NPS Ratings (0-10)** | **CSAT Ratings (1-10)** | **TSR (Success/Fail)** |
| 1 | Danny | 8 | 9 | 100 |
| 2 | John | 7 | 7 | 100 |
| 3 | Eugene | 9 | 8 | 100 |
| 4 | Daniel | 7 | 8 | 100 |
| 5 | Shawn | 10 | 10 | 100 |
|  | | | | |
| **COMPUTERED AVERAGE/SCORE** | | 8.2 | 8.4 | 100% |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TC ID | Test Case Scenario | Desired | | | Actual | | |
| TOT | TSR | UER | TOT | TSR | UER |
| UST01 | Register | 10 | 100% | 0% | 10 | 100% | 0% |
| UST02 | Login | 10 | 100% | 0% | 7 | 100% | 0% |
| UST03 | View Profile | 15 | 100% | 0% | 21 | 100% | 0% |
| UST04 | Update Profile | 10 | 100% | 0% | 10 | 100% | 0% |
| UST05 | Search other users | 10 | 100% | 0% | 8 | 100% | 0% |
| UST06 | View other user’ Public Profile | 15 | 100% | 0% | 20 | 100% | 0% |

TEST CASE ID: UST01 – Register an account

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Participants | TOT  (sec) | UER  (# Errors) | TSR (Success/Fail) | Observations | Feedback |
| 1 | Danny | 35 | 0 | 100 |  |  |
| 2 | John | 30 | 0 | 100 |  |  |
| 3 | Eugene | 28 | 0 | 100 |  |  |
| 4 | Daniel | 35 | 0 | 100 |  |  |
| 5 | Shawn | 22 | 0 | 100 |  |  |
|  | | | | |  |  |
| COMPUTERED AVERAGE/SCORE | | 30 Sec | 0% | 100% |  |  |

TEST CASE ID: UST02 - Log In

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Participants | TOT (sec) | UER  (# Errors) | TSR (Success/Fail) | Observations | Feedback |
| 1 | Danny | 10 | 0 | 100 |  |  |
| 2 | John | 9 | 0 | 100 |  |  |
| 3 | Eugene | 8 | 0 | 100 |  |  |
| 4 | Daniel | 11 | 0 | 100 |  |  |
| 5 | Shawn | 7 | 0 | 100 |  |  |
|  | | | | |  |  |
| COMPUTERED AVERAGE/SCORE | | 9 Sec | 0% | 100% |  |  |
|  |  |

TEST CASE ID: UST03 – View Profile

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Participants | TOT (sec) | UER  (# Errors) | TSR (Success/Fail) | Observations | Feedback |
| 1 | Danny | 10 | 0 | 100 |  |  |
| 2 | John | 9 | 0 | 100 |  |  |
| 3 | Eugene | 8 | 0 | 100 |  |  |
| 4 | Daniel | 11 | 0 | 100 |  |  |
| 5 | Shawn | 7 | 0 | 100 |  |  |
|  | | | | |  |  |
| COMPUTERED AVERAGE/SCORE | | 9 Sec | 0% | 100% |  |  |
|  |  |

TEST CASE ID: UST04 – Update Profile

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Participants | TOT  (sec) | UER  (# Errors) | TSR (Success/Fail) | Observations | Feedback |
| 1 | Danny | 22 | 0 | 100 |  |  |
| 2 | John | 21 | 0 | 100 |  |  |
| 3 | Eugene | 18 | 0 | 100 |  |  |
| 4 | Daniel | 14 | 0 | 100 |  |  |
| 5 | Shawn | 12 | 0 | 100 |  |  |
|  | | | | |  |  |  |
| COMPUTERED AVERAGE/SCORE | | 17.4 Sec | 0% | 100% |  |  |

TEST CASE ID: UST05 – Search other users

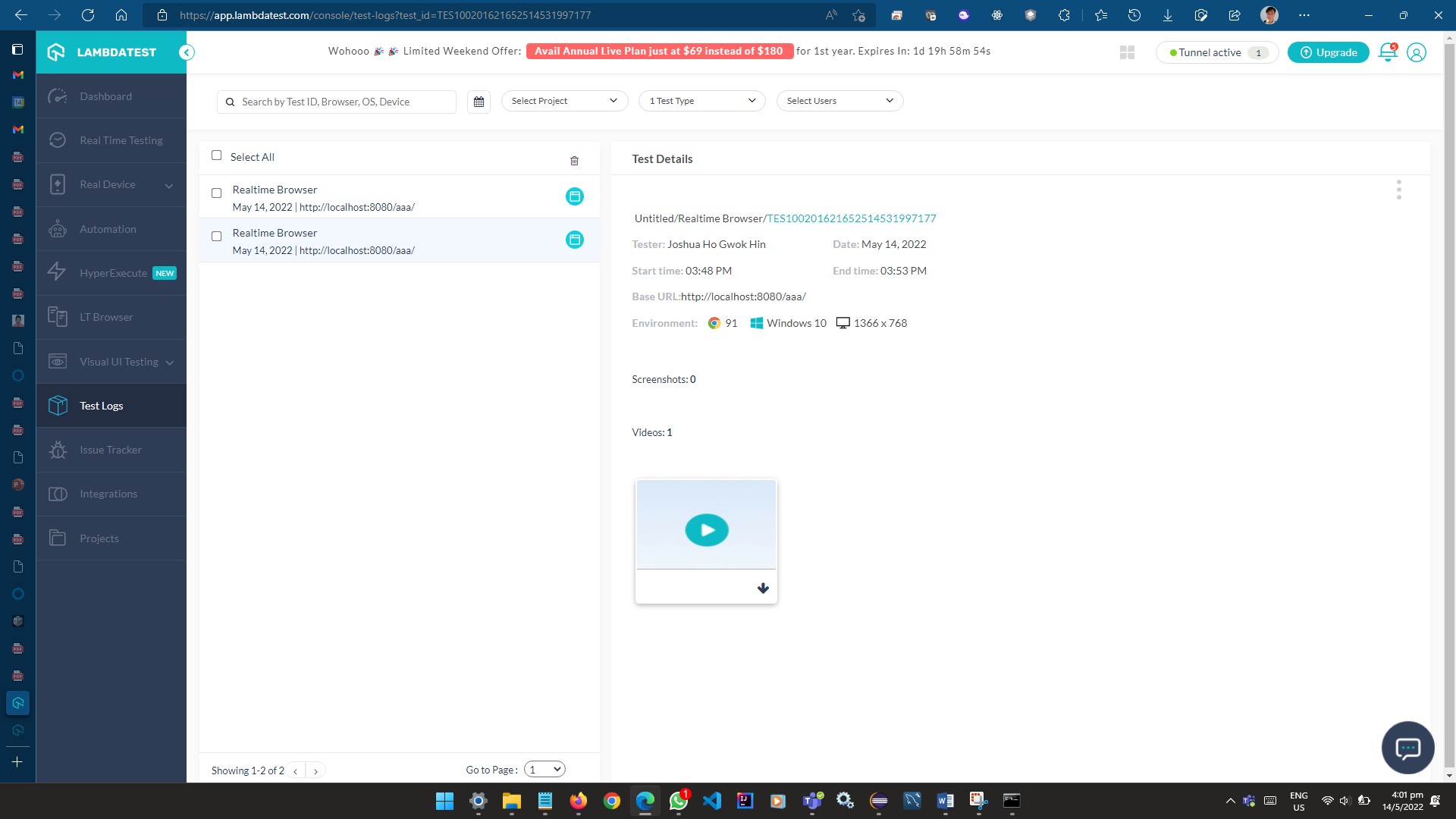
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Participants | TOT  (sec) | UER  (# Errors) | TSR (Success/Fail) | Observations | Feedback |
| 1 | Danny | 25 | 0 | 100 |  |  |
| 2 | John | 30 | 0 | 100 |  |  |
| 3 | Eugene | 20 | 0 | 100 |  |  |
| 4 | Daniel | 35 | 0 | 100 |  |  |
| 5 | Shawn | 22 | 0 | 100 |  |  |
|  | | | | |  |  |
| COMPUTERED AVERAGE/SCORE | | 26.4 Sec | 0% | 100% |  |  |

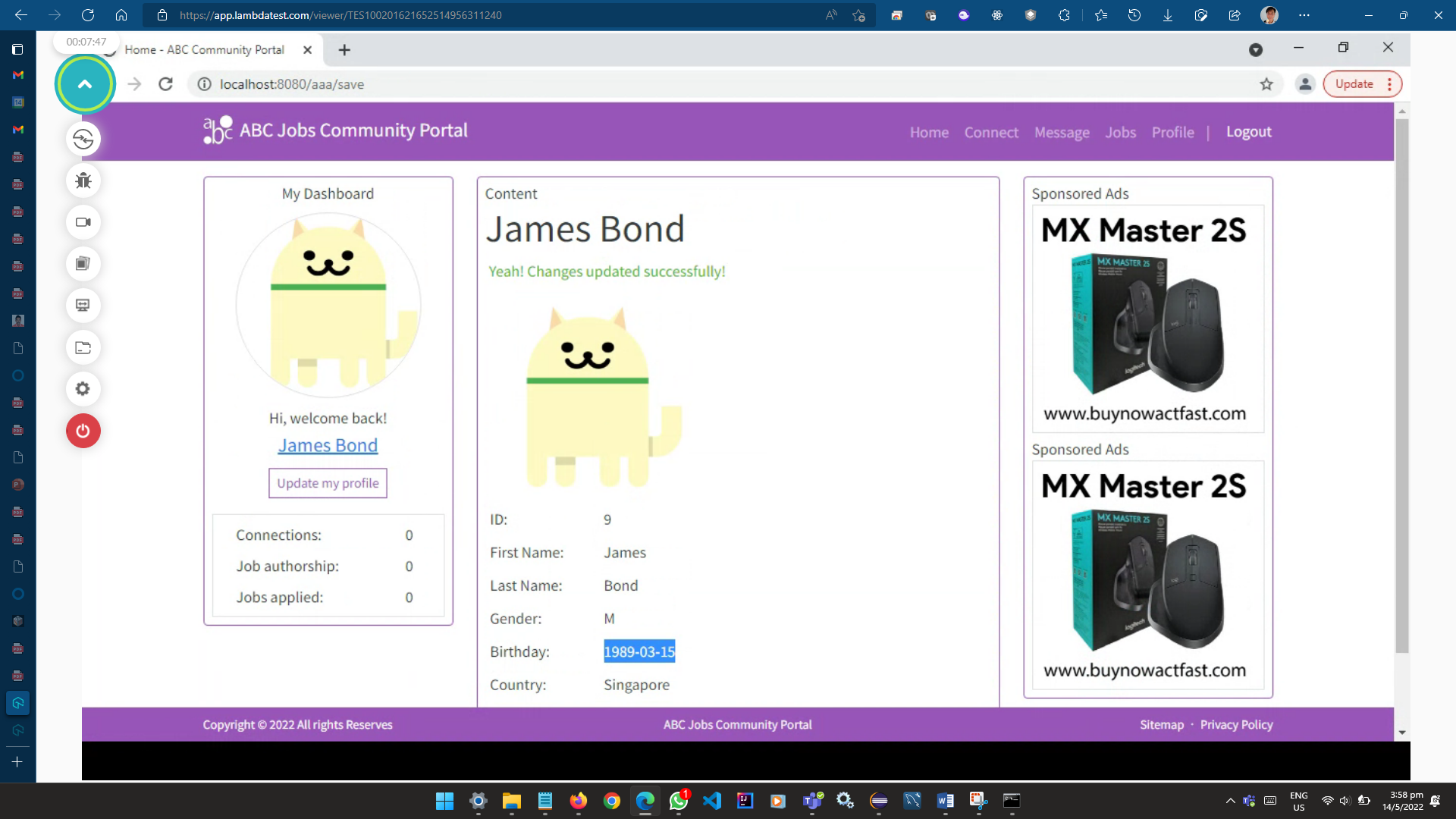
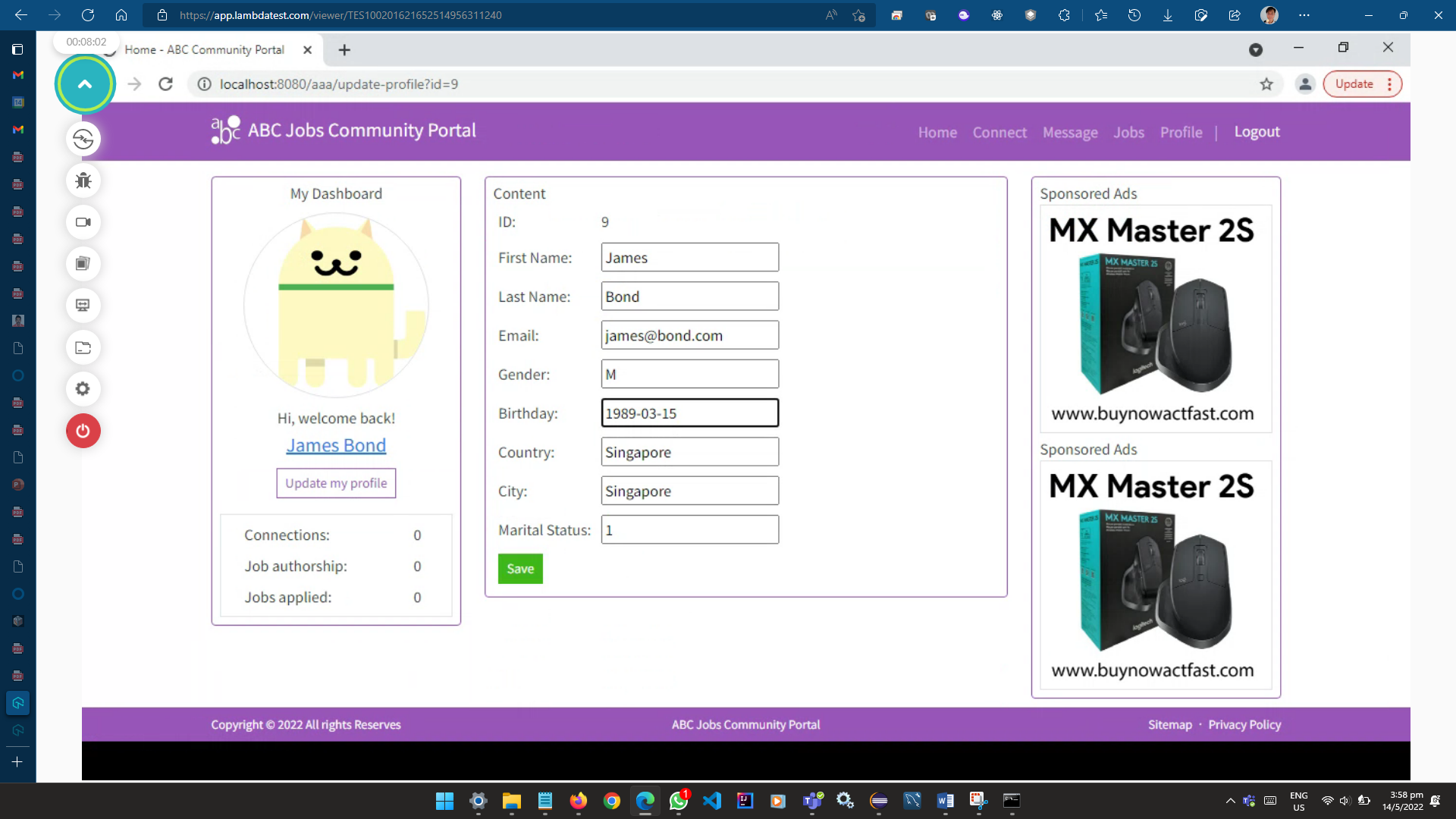
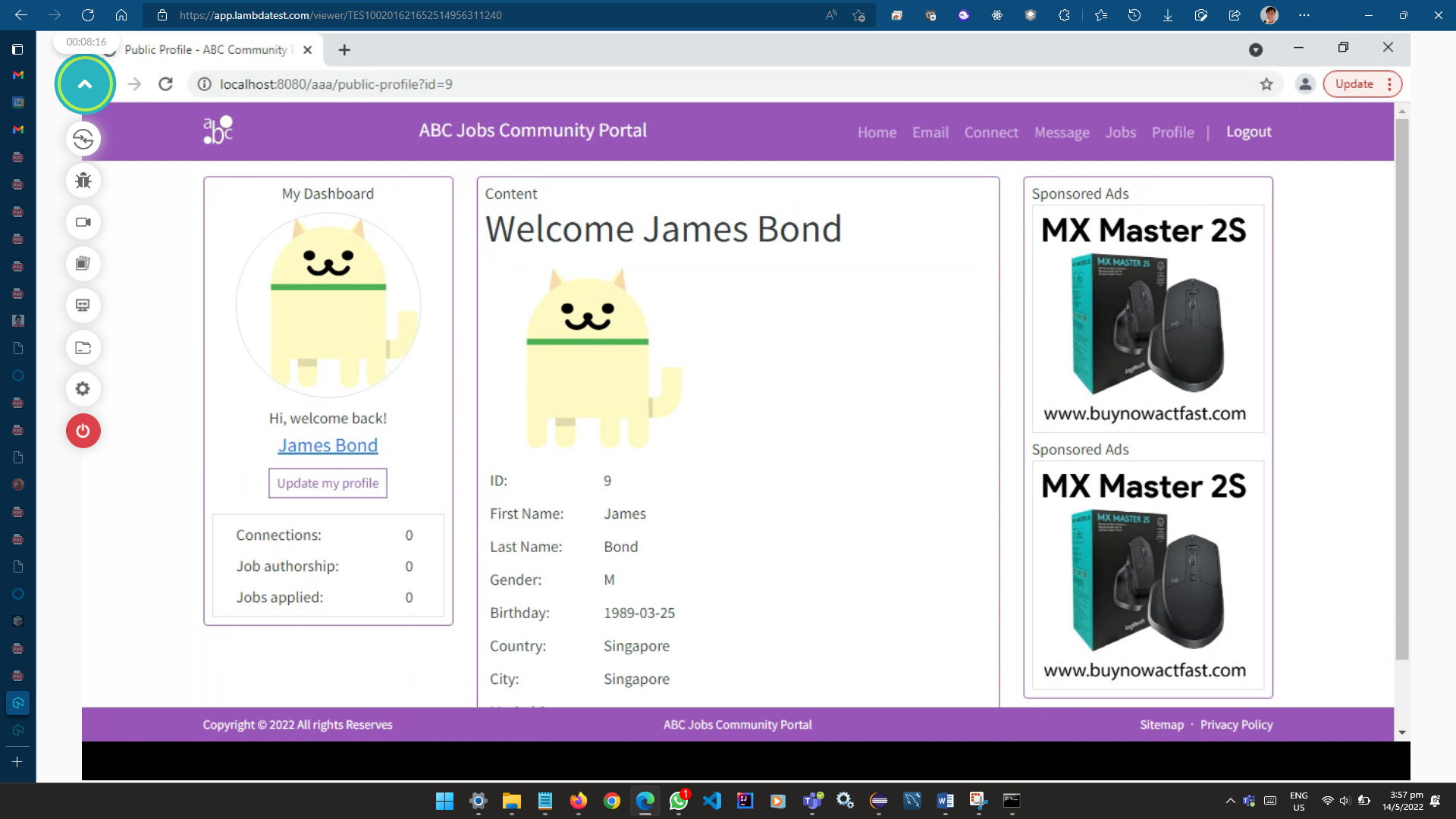
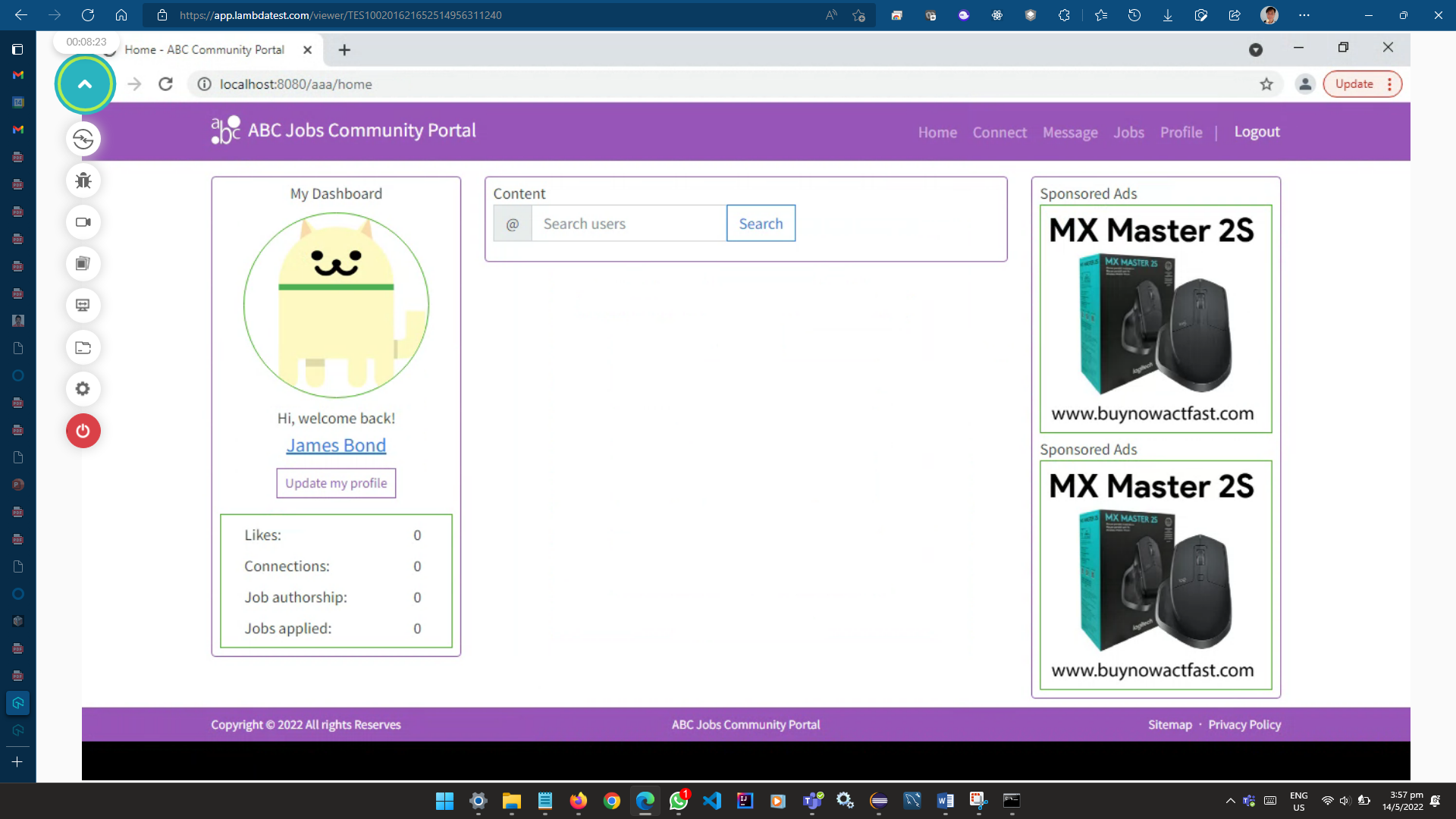
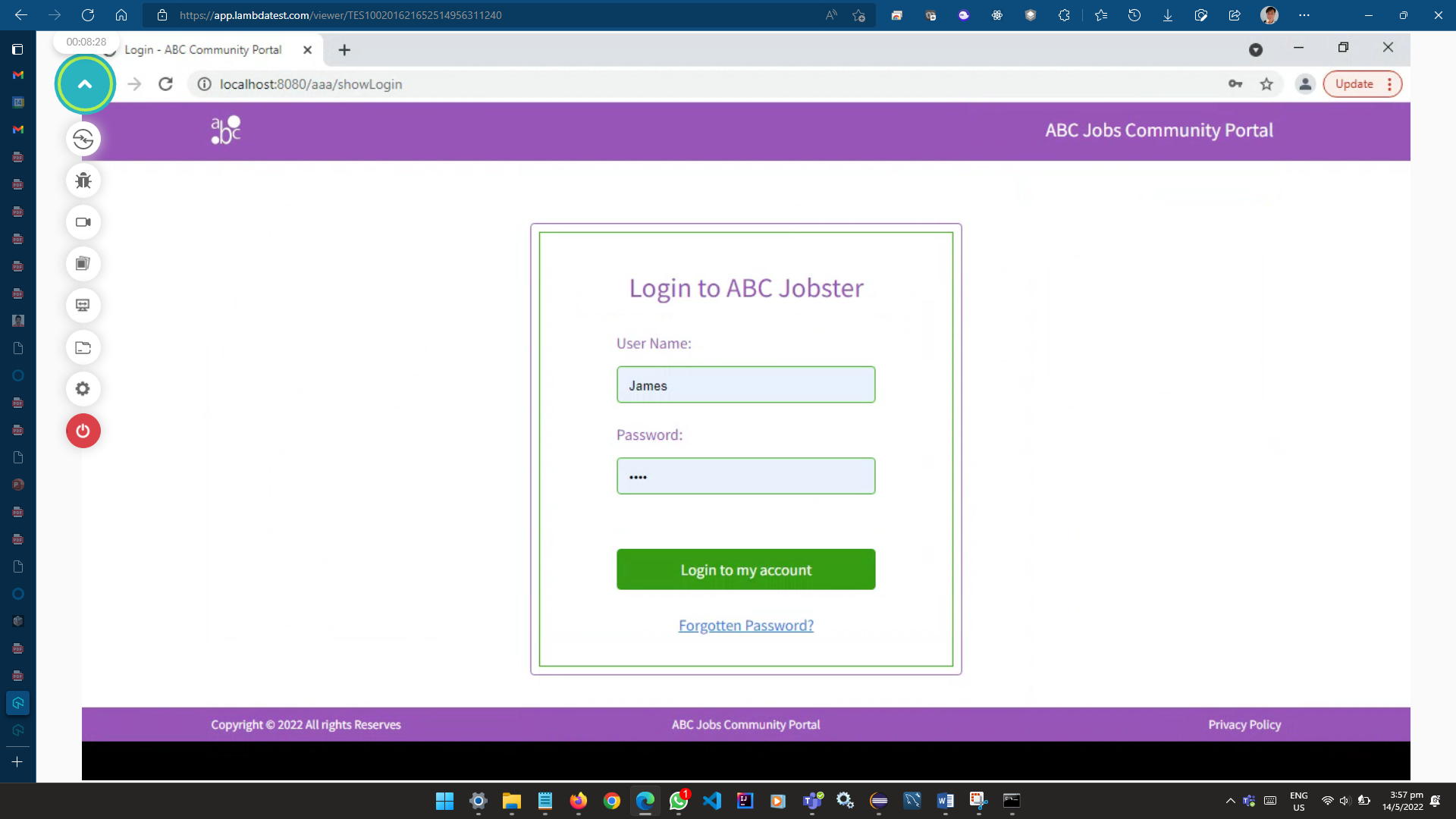
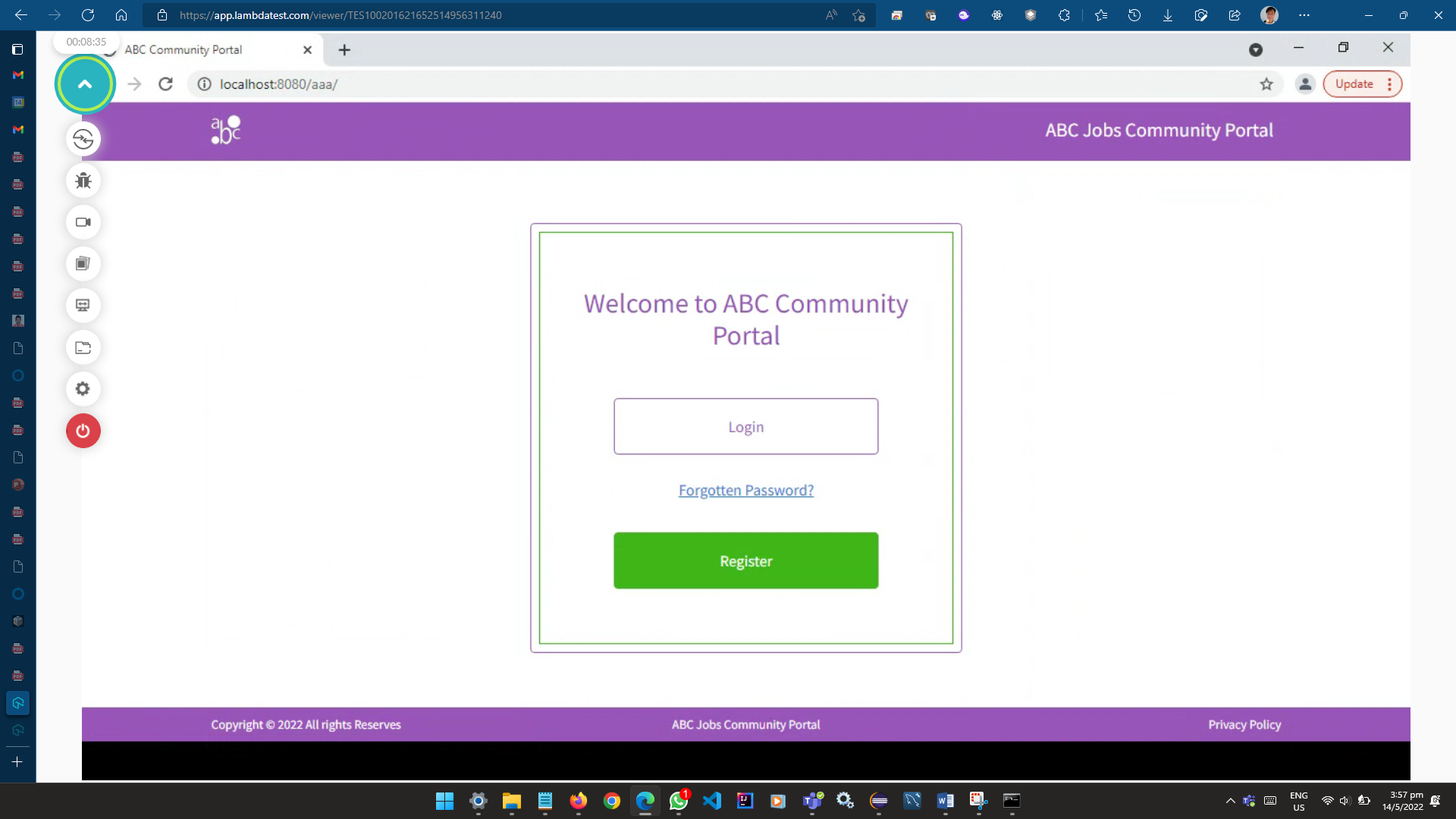
TEST CASE ID: UST06 – View other users’ Public Profile

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Participants | TOT  (sec) | UER  (# Errors) | TSR (Success/Fail) | Observations | Feedback |
| 1 | Danny | 25 | 0 | 100 |  |  |
| 2 | John | 30 | 0 | 100 |  |  |
| 3 | Eugene | 20 | 0 | 100 |  |  |
| 4 | Daniel | 35 | 0 | 100 |  |  |
| 5 | Shawn | 22 | 0 | 100 |  |  |
|  | | | | |  |  |
| COMPUTERED AVERAGE/SCORE | | 26.4 Sec | 0% | 100% |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TEST CASE ID: UNT01 | | | TEST DESCRIPTION: LOGIN | | | | |
| Preconditions:   1. Access to Chrome Browser 2. http://localhost:8080/aaa/showLogin | | | | Test Data   1. Username = James 2. Password = Bond 3. Invalid Username = Banana 4. Invalid Password = Split | | |
| S/no. | Description | Steps | | Expected Results | Actual Result | Status |
| 01 | Valid Login | 1. Enter username as “James”  2. Enter password as “Bond”  3. Click/Tab “Login” | | Login Successful, redirect to home page. | as expected | pass |
| 02 | Invalid Login  Valid Username but wrong password. | 1. Enter username as “James”  2. Enter password as “Split”  3. Click/Tab “Login” | | Login unsuccessful, show error message: “*Username or Password is wrong!!*” | as expected | pass |
| 03 | Invalid Login  Valid Username but no password | 1. Enter username as “James”  2. Skip password field.  3. Click/Tab “Login” | | Login unsuccessful, show error message: “*Username or Password is wrong!!*” | as expected | pass |
| 04 | Invalid Login  Invalid Username. | 1. Enter username as “Banana”  2. Enter password as “Split”  3. Click/Tab “Login” | | Login unsuccessful, show error message: “*Username or Password is wrong!!*” | as expected | pass |
| 05 | Invalid Login  Invalid Username. | 1. Enter username as “Banana”  2. Enter password as “Split”  3. Click/Tab “Login” | | Login unsuccessful, show error message: “*Username or Password is wrong!!*” | as expected | pass |
| TEST CASE ID: UNT02 | | | TEST DESCRIPTION: VIEW PROFILE | | | | |
| Preconditions:   1. Login to valid account 2. http://localhost:8080/aaa/home | | | | Test Data   1. Username = James 2. Password = Bond | | |
| S/no. | Description | Steps | | Expected Results | Actual Result | Status |
| 01 | Display user profile properly | 1. Click on “**James Bond**” in My Dashboard panel. | | Redirected to public-profile.jsp to display public profile with the following details:  ID: 9  First Name: James  Last Name: Bond  Gender: M  Birthday: 1989-12-12  Country: Singapore  City: Singapore  Marital Status: 3 | as expected | pass |
| 02 | Display user profile properly | 1. Click on “**Profile**” from the navigation menu bar at the top of the web page. | | Redirected to public-profile.jsp to display public profile with the following details:  ID: 9  First Name: James  Last Name: Bond  Gender: M  Birthday: 1989-12-12  Country: Singapore  City: Singapore  Marital Status: 3 | as expected | pass |
| TEST CASE ID: UNT03 | | | TEST DESCRIPTION: UPDATE PROFILE | | | | |
| Preconditions:   1. Login to valid account 2. http://localhost:8080/aaa/home | | | | Test Data   1. Username = James 2. Password = Bond 3. First Name = James 4. Last Name = Bond 5. Email = [james@bond.com](mailto:james@bond.com) 6. Gender = M 7. Birthday = 1987-12-31 8. Country= Singapore 9. City = Singapore 10. Marital Status = 1 | | |
| S/no. | Description | Steps | | Expected Results | Actual Result | Status |
| 01 | Validate update profile is properly implemented. | 1. Click “Update my profile” button in My Dashboard panel.  2. Key in Birthday as “1987-12-31”  3. Key in Marital Status as “1”  4. Click the “Save” button. | | Redirected to update-success.jsp to display public profile with the following message & details:  “Yeah! Changes updated successfully!”  ID: 9  First Name: James  Last Name: Bond  Gender: M  Birthday: 1987-12-31  Country: Singapore  City: Singapore  Marital Status: 1 | as expected | pass |
| TEST CASE ID: UNT04 | | | TEST DESCRIPTION: SEARCH OTHER USERS | | | | |
| Preconditions:   1. Login to valid account 2. http://localhost:8080/aaa/home | | | | Test Data   1. Username = James 2. Password = Bond 3. Valid Keyword = Singapore 4. Valid Keyword = Magic 5. Invalid Keyword = Magical | | |
| S/no. | Description | Steps | | Expected Results | Actual Result | Status |
| 01 | Conduct search with valid keyword for Country and/or City. | 1. Click “Search users” input field.  2. Key in “Singapore”  3. Click the “Search” button. | | Display a list of 9 profile having set “Singapore” as Country or City. | as expected | pass |
| 02 | Conduct search with valid keyword First Name and/or Last Name | 1. Click “Search users” input field.  2. Key in “Magic”  3. Click the “Search” button. | | Display 1 profile having First Name set as “Magic”. | as expected | pass |
| 02 | Conduct search with invalid keyword. | 1. Click “Search users” input field.  2. Key in “Magical”  3. Click the “Search” button. | | Display 0 profile. | as expected | pass |

**Cross Browser Testing**





**User Acceptance Testing**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case: UAT01 - Software Programmers | | | |
| Scenario | Disagree | Neutral | Agree |
| The content is easy to understand | 0 | 20% | 80% |
| The visual theme is appropriate (colours, fonts, buttons, etc.) | 0 | 0 | 100% |
| Navigation is intuitive and easy to follow | 0 | 0 | 100% |
| The overall experience is acceptable | 0 | 0 | 100% |

Number of Software Programmers: 5

Conclusion:

4 in 5 of total Software Programmers finds that the content is easy to understand.

5 in 5 of total Software Programmers finds that the visual theme is appropriate.

5 in 5 of total Software Programmers finds that the navigation is intuitive and easy to follow.

5 in 5 of total Software Programmers finds that the overall experience is acceptable.

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case: UAT02 – Network Administrators | | | |
| Scenario | Disagree | Neutral | Agree |
| The content is easy to understand | 0 | 0 | 100% |
| The visual theme is appropriate (colours, fonts, buttons, etc.) | 0 | 0 | 100% |
| Navigation is intuitive and easy to follow | 0 | 0 | 100% |
| The overall experience is acceptable | 0 | 0 | 100% |

Number of Network Administrators: 5

Conclusion:

5 in 5 of total Network Administrators finds that the content is easy to understand.

5 in 5 of total Network Administrators finds that the visual theme is appropriate.

5 in 5 of total Network Administrators finds that the navigation is intuitive and easy to follow.

5 in 5 of total Network Administrators finds that the overall experience is acceptable.

Total Participants: 5+5=10