

American International University-Bangladesh (AIUB)  
Department of Computer Science  
Faculty of Science &Technology (FST)  
Summer 19\_20

Section: A  
Software Quality Assurance and Testing

Automatic Team Members Matching

A Report submitted

By

|  |  |  |
| --- | --- | --- |
| SN | Student Name | Student ID |
| 1 | Rakibul Newaz Sourav | 18-37316-1 |
| 2 | Israt Jahan Dristi | 17-35109-2 |
| 3 | Saiful Islam | 17-35188-2 |
| 4 | Kaniz Fatema | 15-30205-2 |

Under the supervision of

Abhijit Bhowmik

Department of Computer Science

Faculty of Science and Information Technology

American International University-Bangladesh

Software Test Plan

For

Automatic Team Members Matching

Version 1.0 approved

Prepared by

Rakibul Newaz Sourav

Israt Jahan Dristi

Saiful Islam

Kaniz Fatema

American International University-Bangladesh

22 September 2020

Table of Contents

[Revision History 3](#_Toc37271323)

[1. TEST PLAN IDENTIFIER: RS-MTP01.3 4](#_Toc37271324)

[2. REFERENCES 4](#_Toc37271325)

[3. INTRODUCTION 4](#_Toc37271326)

[Background to the Problem 4](#_Toc37271327)

[Solution to the Problem 4](#_Toc37271328)

[4. REQUEIREMNT SPECIFICATION 5](#_Toc37271329)

[4.1 System Features 5](#_Toc37271330)

[4.2 System Quality Attributes 6](#_Toc37271331)

[4.3 System Interface 7](#_Toc37271332)

[4.4 Project Requirements 7](#_Toc37271333)

[5. FEATURES NOT TO BE TESTED 7](#_Toc37271334)

[6. TESTING APPROACH 8](#_Toc37271335)

[6.1 Testing Levels 8](#_Toc37271336)

[6.2 Test Tools 9](#_Toc37271337)

[6.3 Meetings 9](#_Toc37271338)

[7. TEST CASES/TEST ITEMS 10](#_Toc37271339)

[8. ITEM PASS/FAIL CRITERIA 16](#_Toc37271340)

[9. TEST DELIVERABLES 16](#_Toc37271341)

[10. STAFFING AND TRAINING NEEDS 16](#_Toc37271342)

[11. RESPONSIBILITIES 17](#_Toc37271343)

[12. TESTING SCHEDULE 18](#_Toc37271344)

[13. PLANNING RISKS AND CONTINGENCIES 18](#_Toc37271345)

[14. APROVALS 19](#_Toc37271346)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Updated by | Update Comments |
| 0.1 | 2020.9.15 | Rakibul Newaz Sourav | First Draft |
| 0.2 | 2020.9.16 | Rakibul Newaz Sourav | Second draft |
| 0.3 | 2020.9.17 | Rakibul Newaz Sourav | Third draft |
| 0.4 | 2020.9.17 | Saiful Islam | Fourth draft |
| 0.5 | 2020.9.18 | Saiful Islam | Fifth draft |
| 0.6 | 2020.9.18 | Saiful Islam | Sixth draft |
| 0.7 | 2020.9.18 | Israt Jahan Dristi | Seventh draft |
| 0.8 | 2020.9.19 | Israt Jahan Dristi | Eighth draft |
| 0.9 | 2020.9.20 | Kaniz Fatema | Ninth draft |
| 1.0 | 2020.9.21 | Kaniz Fatema | Tenth draft |

# TEST PLAN IDENTIFIER:

Automatic Team Members Matching

REFERENCES

* Psychological Testing: Theory and Practice by Collin Cooper

# INTRODUCTION

## Background to the Problem

In our daily life, when we work as a team for a particular task. There are lots of teams working into the same project. Making a great team is a key feature for project success. We are not machines, we act based on our mentality. That's why making a team where everybody's mentality is the same, boosts our working experience. When it comes to choosing the same mentality people there is only one way by taking a psychological test. We build our website based on this psychological test. Our test is a simple multiple-choice quiz test. We have our different types of quiz questions. After finishing every quiz people get a score. It measures their mentality. Same score people are going to the same team. Our quiz questions are randomly selected for a participant. There is no wrong or right answer for each question. People can choose any answer. Based on that answer our system will select the next question automatically. Every question has .5 minute to answer. Participants cannot skip any question. We store every participant score to their account.

## Solution to the Problem

Most of the quiz tests are based on individual psychological tests online. Also, you can give those tests for your own good. As an example, if somebody wants to buy a car but cannot decide what color he/she wants. Giving a psychological test decides the percentage for each choosing color. This is the main difference between our website and other websites. We work on a particular team or group on our website. On our website we try to measure physiological scores for each team member. After this test they find out their comfortness to each other based on their psychological test score. If they are at the same comfortness level then they can work together or if not, they are not suitable for working together.

# REQUEIREMNT SPECIFICATION

## System Features

* **User Features**

1. Sign up to our website to make their own account.
2. Participants can logout from their account.
3. Participants can see our blog post.
4. Participants can comment on our blog post.
5. Participants can attend any quiz test which we provide.
6. Participants can edit/update their account.
7. Project manager can create room.
8. Participants can contact us using contact form

* **Moderator Features**

1. Moderator can login to their account.
2. Moderators can edit/update their account.
3. Moderators can replay participant messages.
4. Moderators can edit blog posts.
5. Moderator can publish a new post.
6. Moderator can make quiz questions paper.
7. Moderator can see participant information.

* **Admin Features**

1. Admin can login to their account.

1.1 The software shall allow users to login with their given username and password

1.2 If the username and/or password has been inserted wrong for more than three times, the random verification code will be generated by the system to retry login.

1.3 If the number of login attempt exceed its limit (5 times), the system shall block the user account login for one hour *[optional function]*

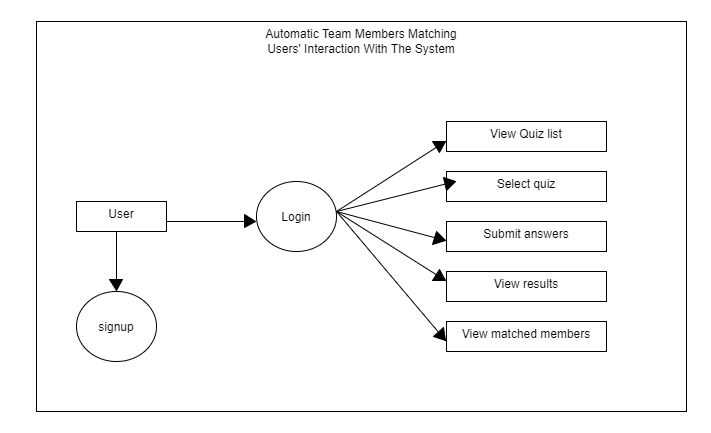
Priority Level: High  
Precondition: user have valid user id and password

1. Admin can edit/update their account.
2. Admin can do all things which moderators can do.
3. Admin can add another moderator.
4. Admin can see moderator information.
5. Admin can delete any question set

## System Quality Attributes

* **Usability:** This can be measured in terms of ease of use. The application should be user-friendly. It should be easy to learn. Navigation should be simple. The system must be: Easy to use for input preparation, operation, and interpretation of the output. Provide consistent user interface standards or conventions with our other frequently used systems. Easy for new or infrequent users to learn to use the system.
* **Reliability:** Measure if the software is reliable enough to sustain in any condition. Should give consistently correct results. Software reliability will be measured in terms of working of the project under different working environments and different conditions.
* **Maintainability:** Different versions of the software should be easy to maintain. For development it will be easy to add code to the existing system, and should be easy to upgrade for new features and new technologies from time to time. Maintenance must be cost-effective and easy. The system should be easy to maintain and correct defects or making a change in the software.
* **Portability:** It should cost low in terms of costing issues, easy to migrate in terms of technical issues.
* **Correctness:** The application should be correct in terms of its functionality, calculations used internally and the navigation should be correct. This means the application will adhere to functional requirements.
* **Efficiency:** Major system quality attribute. Measured in terms of time required to complete any task given to the system. The application will have large scale in efficiency.
* **Integrity or Security:** Integrity comes with security. System integrity or security should be sufficient to prevent unauthorized access to system functions, preventing information loss, ensure that the software is protected from virus infection, and protecting the privacy of data entered into the system.
* **Testability:** The system should be easy to test and find defects. If required it should be easy to divide into different modules for testing.
* **Flexibility:** Should be flexible enough to modify. Adaptable to other products with which it needs interaction. Should be easy to interface with other standard 3rd party components.
* **Reusability:** Software reuse is a good cost-efficient and time-saving development way. Different code libraries classes should be generic enough to use easily in different application modules. Dividing application into different modules so that modules can be reused across the application.

## System Interface



## Project Requirements

* Time: Total development time is 6 months
* Budget: Total budget 1,30,000 BDT
* Resource: Very limited resource
* Quality: Low quality risk for short time

# FEATURES NOT TO BE TESTED

The following is a list of the areas that will not be specifically addressed. All testing in these areas will be indirect as a result of other testing efforts. For example:

**Multi Language System**: In our web application there are two types of language one is English another one is Bangla. But most of our participants select our default language which is English. Very few participants actually choose the second option.

**Quiz Post Reaction with Emoji:** We provide multiple emoji to react to our quiz posts. But most of the time our participants like to comment and not react.

**Moderator Theme change option:** Moderator can change theme or fronts size to our website only just clicking some button. But it is not a necessary option to test in the first phase. Also, it does not create any impact on our daily participant. Because every theme or font size is already much comfortable for our participant.

**Auto Location Select Function:** Most of the time participants select their location by scrolling our location check box. But when participant hit the location box. Automatic location will be suggested, if participants want, they can select that by hitting that suggested location

# TESTING APPROACH

1. **Unit Testing:** Each individual piece of code will be tested. Ex: each class or method. In this step, we will implement “White Box Testing” technique.
2. **Integration Testing:** After that, we will test the integrations of many units together. We will make sure code works when they are put together, including dependencies, databases and libraries. In this step, we will follow the “Bottom-up Integration” technique.
3. **System Testing:** We will test the integrated system to verify that it meets specified requirements. At this level, we will follow “Black Box Testing” technique.
4. **Acceptance Testing:** Finally, we will test that the website works the way a user/customer expects the application to work. We will ensure that the functionality meets business requirements. At this level, we will follow the “Black Box Testing” technique. After that, we may run unit tests again. This is regression testing to ensure that further changes have not broken any units that were already tested. We will run unit tests again and again for regression testing.

## Testing Levels

* The testing for this project will consist of Unit, System/Integration (combined) and Acceptance test levels. It is hoped that there will be at least one full time independent test person for system/integration testing. However, with the budget constraints and timeline established; most testing will be done by the test manager with the development teams’ participation.
* UNIT Testing will be done by the developer and will be approved by the development team leader. Proof of unit testing (test case list, sample output, data printouts, defect information) must be provided by the programmer to the team leader before unit testing will be accepted and passed on to the test person. All unit test information will also be provided to the test person.
* SYSTEM/INTEGRATION Testing will be performed by the test manager and development team leader with assistance from the individual developers as required. No specific test tools are available for this project. Programs will enter into System/Integration test after all critical defects have been corrected. A program may have up to two Major defects as long as they do not impede testing of the program (I.E. there is a work around for the error).
* ACCEPTANCE Testing will be performed by the actual end users with the assistance of the test manager and development team leader. The acceptance test will be done in parallel with the existing manual ZIP/FAX process for a period of one month after completion of the System/Integration test process.

## Test Tools

1. Eclipse IDE (For writing testing codes and execution)
2. Selenium (Testing framework)
3. Web Browser (Google Chrome)

## Meetings

The test team will meet once every two weeks to evaluate progress to date and to identify error trends and problems as early as possible. The test team leader will meet with development and the project manager once every two weeks as well. These two meetings will be scheduled on different weeks. Additional meetings can be called as required for emergency situations.

# TEST CASES/TEST ITEMS

## Signup

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Project Name:** Automatic Team Member Matching | | | **Test Designed by:** Rakibul Newaz Sourav | | |
| **Test Case ID**: Signup\_01 | | | **Test Designed date:** 15-9-20 | | |
| **Test Priority (Low, Medium, High):** High | | | **Test Executed by**: Rakibul Newaz Sourav | | |
| **Module Name:** Sign Up | | | **Test Execution date:** 20-9-20 | | |
| **Test Title:** Verifying sign up with valid username, email and password | | |  | | |
| **Description:** Test the website sign up page | | |  | | |
| **Precondition (If any):** Need a valid email id | | | | | |
| **Test Steps** | **Test Data** | **Expected Results** | | **Actual Results** | **Status (Pass/Fail)** |
| 1. Go to the site.  2. Click the “sign up” button 3. Enter user name.  4. Enter email id.  5. Enter password.  6. Click “sign up” button | Username: sourav  Email: sourav@gmail.com Password: 12345 | Users should  Signup to the  website | | As expected, | Pass |
| **Post Condition:** User is validated with database and successfully register an account. The account details are stored in the database | | | | | |

## Login

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Project Name:** Automatic Team Member Matching | | | **Test Designed by:** Rakibul Newaz Sourav | | |
| **Test Case ID:** Login\_02 | | | **Test Designed date:** 15-9-20 | | |
| **Test Priority (Low, Medium, High):** High | | | **Test Executed by**: Rakibul Newaz Sourav | | |
| **Module Name:** login | | | **Test Execution date:** 20-9-20 | | |
| **Test Title:** Verifying login with valid username, email and password | | |  | | |
| **Description:** Test the website login option | | |  | | |
| **Precondition (If any):** Need account on this website | | | | | |
| **Test Steps** | **Test Data** | **Expected Results** | | **Actual Results** | **Status (Pass/Fail)** |
| 1. Click the “Login” button 2. Enter username/email. 3. Enter password.  4. Click the “Login” button | Username: sourav  Email: sourav@gmail.com Password: 12345 | Users should  Logged in to the  website | | As expected, | Pass |
| **Post Condition:** User’s personal account is shown into the system | | | | | |

## Logout

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Project Name:** Automatic Team Member Matching | | | **Test Designed by:** Rakibul Newaz Sourav | | |
| **Test Case ID:** Logout\_03 | | | **Test Designed date:** 15-9-20 | | |
| **Test Priority (Low, Medium, High):** High | | | **Test Executed by**: Rakibul Newaz Sourav | | |
| **Module Name:** logout | | | **Test Execution date:** 20-9-20 | | |
| **Test Title:** Verifying logout option | | |  | | |
| **Description:** Test the website logout option | | |  | | |
| **Precondition (If any):** 1. Need account on this website  2. Need to be logged in | | | | | |
| **Test Steps** | **Test Data** | **Expected Results** | | **Actual Results** | **Status (Pass/Fail)** |
| 1. Click the “Logout” button |  | Successful logout | | As expected, | Pass |
| **Post Condition:** User goes back to the main page | | | | | |

## Add Quiz

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Project Name:** Automatic Team Member Matching | | | **Test Designed by:** Rakibul Newaz Sourav | | |
| **Test Case ID:** add\_quiz\_04 | | | **Test Designed date:** 15-9-20 | | |
| **Test Priority (Low, Medium, High):** High | | | **Test Executed by**: Rakibul Newaz Sourav | | |
| **Module Name:** Quiz | | | **Test Execution date:** 20-9-20 | | |
| **Test Title:** Add new quiz | | |  | | |
| **Description:** Test the adding new quiz option | | |  | | |
| **Precondition (If any):** 1. Need to be admin  2. Need to be logged in | | | | | |
| **Test Steps** | **Test Data** | **Expected Results** | | **Actual Results** | **Status (Pass/Fail)** |
| 1. click ‘’login’’  2. enter email  3. enter password  4. click “login”  5. click “add quiz”  6. Enter quiz Name  7. Select quiz type  8. click “add quiz” | **Quiz Title:**  Sample quiz  **Quiz type:**  Demo | Added new quiz | | As expected, | Pass |
| **Post Condition:** Admin redirect to the same dashboard page with a success message | | | | | |

## Delete Quiz

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Project Name:** Automatic Team Member Matching | | | **Test Designed by:** Rakibul Newaz Sourav | | |
| **Test Case ID:** update\_quiz\_05 | | | **Test Designed date:** 15-9-20 | | |
| **Test Priority (Low, Medium, High):** High | | | **Test Executed by**: Rakibul Newaz Sourav | | |
| **Module Name:** Quiz | | | **Test Execution date:** 20-9-20 | | |
| **Test Title:** delete quiz | | |  | | |
| **Description:** Test the deleting quiz option | | |  | | |
| **Precondition (If any):** 1. Need to be admin  2. Need to be logged in | | | | | |
| **Test Steps** | **Test Data** | **Expected Results** | | **Actual Results** | **Status (Pass/Fail)** |
| 1. click ‘’login’’  2. enter email  3. enter password  4. click “login”  5. click “delete quiz”  6. Enter quiz Name  7. Select quiz  8. click “delete” | **Selected quiz:**  Sample quiz | Quiz deleted | | As expected, | Pass |
| **Post Condition:** Admin redirect to the same dashboard page with a success message | | | | | |

## Add Question

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Project Name:** Automatic Team Member Matching | | | **Test Designed by:** Rakibul Newaz Sourav | | |
| **Test Case ID:** add\_question\_06 | | | **Test Designed date:** 15-9-20 | | |
| **Test Priority (Low, Medium, High):** High | | | **Test Executed by**: Rakibul Newaz Sourav | | |
| **Module Name:** Quiz | | | **Test Execution date:** 20-9-20 | | |
| **Test Title:** add question | | |  | | |
| **Description:** Test the adding new question option | | |  | | |
| **Precondition (If any):** 1. Need to be admin  2. Need to be logged in | | | | | |
| **Test Steps** | **Test Data** | **Expected Results** | | **Actual Results** | **Status (Pass/Fail)** |
| 1. click ‘’login’’  2. enter email  3. enter password  4. click “login”  5. click “add question”  6. Select quiz  7. Enter a question  8. enter option A  9. enter option B  10. enter option C  11. enter option D  8. click “add question” | **Selected quiz:**  Sample quiz  **Quiz question:**  Do you like software testing?  **Option A:**  Yes  **Option B:**  No  **Option C:**  Sometimes | Question added | | As expected, | Pass |
| **Post Condition:** Admin redirect to the same dashboard page with a success message | | | | | |

# ITEM PASS/FAIL CRITERIA

If team members report that there are 40% of test cases failed, the project will suspend testing until the development team fixes all the failed cases

# TEST DELIVERABLES

* Test plan
* Test results documents
* Test summary
* Errors
* Bug report
* Defect reports
* Test logs

# STAFFING AND TRAINING NEEDS

**Key areas to set up in Staffing and Training Needs:**

* System and applications.
* Test data.
* Front-end running environment.
* Browser.
* Hardware includes Server Operating systems.
* Network.

**Process of Staffing and Training Needs Setup:**

* System Admins
* Developers
* Testers
* Moderator

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Role | Responsibility | Required Staff |
| Rakibul Newaz Sourav | Project Manager | All | 1 |
| Saiful Islam | Programmer | Coding | 1 |
| Rakibul Newaz Sourav | Quality Analyst | All | 1 |
| Saiful Islam | Test Engineer | Testing | 1 |
| Kaniz Fatema | Tester | Testing | 1 |
| Kaniz Fatema | Programmer | Front end develop | 1 |

# RESPONSIBILITIES

|  |  |  |
| --- | --- | --- |
| Name | Role | Responsibilities |
| Rakibul Newaz Sourav | Test lead | 1. Defining the testing activities  All responsibilities of test planning  2. To check if the team has all the necessary resources to execute the testing activities.  3. Prepare the report of testing activities.  4. Updating project managers regularly about the progress of testing activities |
| Saiful Islam | Test Engineer | 1. Develop test cases and prioritize testing activities.  2. Execute all the test cases and report defects.  3. Inform the test lead about what all resources will be required for software testing. |
| Kaniz Fatema | Software tester | 1. Designing testing scenarios for testing.  2. Perform the testing  3. Creating testing documentation and does testing related works |
| Israt Jahan Dristi | Software tester | 1. Creation of test designs, test processes, test cases and test data.  2. Carry out testing as per the defined procedures.  3.Prepare all reports related to software testing carried out. |

# TESTING SCHEDULE

|  |  |  |
| --- | --- | --- |
| Task Name | Duration | Responsible |
| Documentation | 1 week | Test Lead |
| Design | 1 week | Software Tester |
| Test Plan | 1 week | Test Engineer |
| Unit Testing | 1 week | Software Tester |
| Integration Testing | 4 days | Software Tester |
| System Testing | 1 week | Software Tester |
| Acceptance Testing | 3 days | Software Tester |
| Project Completion | 1 week | Test Engineer |
| Feedback | 5 days | Test Lead |



# PLANNING RISKS AND CONTINGENCIES

|  |  |  |  |
| --- | --- | --- | --- |
| Risk | Probability | Impact | Mitigation |
| Error in function | Medium | Medium | Test the web app frequently and maintain daily backup. |
| Wrong sql command for special data | Medium | high | Maintain security check and backup |
| Loss of encrypted data(password) | Medium | high | Maintain security check and backup |
| User’s account hacking attempt | High | High | Restrict users after 3 unsuccessful login attempts |

# APROVALS

|  |  |
| --- | --- |
| **Position** | **Assigned** |
| Project Sponsor | Rakibul Newaz sourav |
| Development Management | Rakibul Newaz Sourav |
| EDI Project Manager | Saiful Islam |
| RS Test Manager | Saiful Islam |
| RS Development Team Manager | Israt Jahan Dristi |
| Reassigned Sales | Kaniz Fatema |
| Order Entry EDI Team Manager | Kaniz Fatema |