|  |  |
| --- | --- |
| Test Plan – MODULAR STUDY APPLICATION | Team 1: Kory Bennett / Brock Allton / Dylan Sawyer / Justin Casey / Dan Page / Cale Ward  UMUC CMSC 495 6380 27 January 2020 |

TABLE OF CONTENTS

|  |  |
| --- | --- |
| Introduction | 3 |
| Revision History | 3 |
| Project Details | 4 |
| Test Cases | 4 |
|  |  |
|  |  |
|  |  |

Introduction:

The Modular Study System is a program that can be utilized by students at any level from grade school through college. The purpose is to allow the user, a student, to load practice test questions on each subject and then test themselves on those questions. The program will store the questions that are loaded, the questions that are asked, the answer given for those questions, and keep track of how many times the questions were answered correctly or incorrectly.

Revision History:

|  |  |  |  |
| --- | --- | --- | --- |
| Revision Number | Date | Author | Description |
| 1 | 27 Jan 2020 | K. Bennett | Initial Draft |
| 2 | 28 Jan 2020 | B. Allton | Added Items to the test plan |
| 3 | 29 Jan 2020 | K. Bennett | Added Content Intro |
| 4 | 29 Jan 2020 | C. Ward | Added Test Cases |
| 5 | 31 Jan 2020 | D. Sawyer | Added Test Cases |
|  |  |  |  |

Project Details:

|  |  |
| --- | --- |
| Components | Details |
| Project Name | Modular Study Application |
| Authors | Kory Bennett, Brock Allton, Cale Ward |
| Testers |  |
| Testing Environment | Windows 10 Pro Build 1809 – 1909 Ubuntu 16.04 |
| Version | 1 |
| Description |  |

Test Cases:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Number | Action | Expected Results | Actual Results | Pass? |
| 1 | Double click on icon to open application | Application should open to the interface GUI, the top will have the course choices, a field to choose how many questions to answer, and a submit button. The bottom field will contain a field to display the questions and possible answers, radio buttons to select an answer, and a submit button. |  |  |
| 2 | Choose History from the drop-down buttons to choose from. And type in 20 for the number of questions to be asked, click submit button | The field at the bottom should display the first question from the history database to be asked. |  |  |
| 3 | Choose either A, B, C or D radio button to answer the question, click submit | The next questions should be displayed in the field to be answered. |  |  |
| 4 | Answer the requested amount of answers according to process in step 3. | The results of how many questions answered correctly should be displayed on the bottom panel, the text field should be empty, and the drop-down menu for the course to quiz on should reset to the default. |  |  |
| 5 | Select Math from the drop-down box. Select 100 questions to ask. | Error message should pop up stating that the number of questions to be asked exceeds the total number of available questions. |  |  |
| 6 | Keep Math selected, choose 20 questions to answer and click submit. | Math questions and multiple-choice answers should be displayed in text field in bottom panel. |  |  |
| 7 | Answer all 20 questions, and click submit. | Again, the results should be displayed in the bottom panel and the drop-down field at the top reset. |  |  |
| 8 | Select Geography from the drop down and 20 questions to answer | Geography questions and multiple-choice answers should display in the text field. |  |  |
| 9 | Answer all 20 questions, and click submit | The results should be displayed in the bottom panel and the drop-down box reset. |  |  |
| 10 | Select History in the drop-down box, and 20 questions to answer, and go through the questions. | History questions should display in the text field, however, the questions should be different from the first time. |  |  |
| 11 | Click on the Exit button or the x at the top right of the application | Application closes. |  |  |
| 12 | User is in the middle of the test, switches focus to another application, then back to the test. | Test should remain active with no effect on scores or operability. |  |  |
| 13 | Application is minimized or resized. | Program should not crash and maintain full functionality in both cases of resizing or minimizing the window. |  |  |

Test Case 2:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Action | Expected Results | Actual Results | Pass? |
| 1 | Double click on icon to open application | Application opens and the user is prompted for a subject from a drop down menu and the number of questions they would like to be tested on |  |  |
| 2 | Select a subject from the drop down menu | The menu drops down and allows a user to pick |  |  |
| 3 | Type in the “5” in the questions to be asked slot | The number is displayed in a text field for the user to see |  |  |
| 4 | Click start | The current window closes and a new window opens. The new window has a question about the subject displayed and 4 options to choose an aswer from |  |  |
| 5 | Select an answer and click submit | Only one answer is selected and after submit is selected the application brings up the next question |  |  |
| 6 | Repeat step 5, 4 more times | The same outcome as step 5 is expected until the last question is submitted. The application will close and another window will show the results |  |  |
| 7 | The user selects restart | the application closes and a new window with the menu to select a subject opens to begin the test |  |  |
| 8 | The user closes the application by the “x” in the top right corner | The application closes |  |  |