|  |  |
| --- | --- |
| User Guide – MODULAR STUDY APPLICATION | Team 1: Kory Bennett / Brock Allton / Dylan Sawyer / Justin Casey / Dan Page / Cale Ward |

TABLE OF CONTENTS

|  |  |
| --- | --- |
| Revision History | 3 |
| Introduction | 3 |
| Maria Db Backend | 3 |
| Gui Front End | 3 |
| System Requirements | 4 |
| Getting Started | 4 |
|  |  |

Revision History:

|  |  |  |  |
| --- | --- | --- | --- |
| Revision Number | Date | Author | Description |
| 1 | 27 Jan 2020 | K. Bennett | Initial Draft |
| 2 | 31 Jan 2020 | D. Sawyer | Second Draft |
| 3 | 02 Feb 2020 | k. Bennett | Third Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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.

Maria Db Backend:

Maria Db, a SQL database variant will be responsible for housing the content of the Modular Study System. Containing a variety of subjects and questions within each subject. The content of the database will be provided to the user via a “Middle Man” java processor into the GUI Frontend.

GUI Frontend:

The Graphical User Interface will serve as the primary interaction with the user. The GUI is java based and will handle the input/output (IO) functions of the project. From here the user can select study subjects, answer questions and receive test scores.

System Features:

* Multiple study subject selection
* Multiple choice testing
* End of test score feedback
* Randomized questions
* Variable test lengths

Getting Started:

**System Requirements:**

* Intel Core i5 or equivalent
* 2 GB memory (32-bit) or 4 GB memory (64-bit)
* 100 GB Storage Capacity
* Windows 7 or higher

**Selecting a Test:**

Upon executing the program, the initial state will open a Jframe containing the controls to select a test subject. Navigate through the list of subjects, selecting the desired test. Next select the number of questions desired for the test. The number of questions available will appear next to the number entry.

Note: If more questions are requested than are available, the user will be notified.

**Taking a Test:**

Once the start button is clicked after selecting a test, the questions will be displayed in random order to the user. The question format is multiple choice with four answers and only one correct answer. Click the correct answer and the submit button. The next question will appear until the number of requested questions have been answered.

**Scoring a Test:**

Once the final question has been answered, a final submit button will appear. Pressing this button will display the test results as a percentage of corrected answers divided by the total questions. Once finished the user may return to the main menu where another test of the same or different subject can be selected.

Note: Scores of one test do not affect the score of any other test.