School Application

Introduction

The scope of this document is to detail the process of development and testing of the program which has been requested by schools.

The requirements of the application have been identified as:

* Menu system to enable users to easily navigate the system
* Help command to be available at all times to the user
* Mathematical calculations
* Store values of mathematical calculations
* Limited functionality for ease of use to the user
* Classes with update and accept so that the user can see which classes they have for the day
* Spell checker to offer help on difficult words
* Thesaurus to offer alternative words to the user

Games:

* Higher or lower game
* Hangman game

Additional modules:

* Maths Test
* Riddle of the day

Design

The program should be built with simplicity in mind so that future work on the project is easier to implement. For this the use of module files for each module will be used so that each module can be worked on independently to reduce the risk to other functions within the system. The modules will be named similar to that of the requirements above.

Implementation Notes

Variables and modules have been named using camelCase style. The produced file names are:

Testing Plan

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Input | Expected | Result |
| 1 |  |  | As expected |
| 2 |  |  | As expected |
| 3 |  |  | As expected |
| 4 |  |  | As expected |
| 5 |  |  | As expected |
| 6 |  |  | As expected |
| 7 |  |  | As expected |
| 8 |  |  | As expected |
| 9 |  |  | As expected |
| 10 |  |  | As expected |
| 11 |  |  | As expected |
| 12 |  |  | As expected |
| 13 |  |  | As expected |
| 14 |  |  | As expected |

Conclusion

References

Appendixes