CHAPTER 6

TESTING

6.1 The component/sub-system tests

6.1.1 Unit Testing: This is the lowest level of testing that is conducted to remove syntax & logic errors from a single unit. Individual components are tested to ensure that they operate correctly. Each component is tested independently, without other system components.

INPUT*:* search a character

OUTPUT*:* displays output.

6.1.2 Module testing: A module is a collection of dependent components such as an object class, an abstract data type or some looser collection of procedures & functions. A module encapsulates related components, so can be tested without other system modules.

INPUT: search exact word.

OUTPUT*:* displayed right output.

6.1.3 Sub-System testing: This phase involves testing collections of modules, which have been integrated into sub-systems. This tests for problems that arise from component interactions. This testing should begin as soon as usable versions of some of the system components are available.

INPUT*:* search multiple words.

OUTPUT*:* supports only single word search.

6.2 System test

The sub-systems are integrated to make up the system. The system as a complete entity is tested over here. This process is concerned with finding errors that result from unanticipated interactions between sub-systems. It is also concerned with validating that the system meets its functional & non-functional requirements & testing the emergent system properties.

***INPUT**:* Run the software as a whole by inserting data into different sub-systems which are combined together.

***OUTPUT**:* displayed in the Command line interface(CLI).