CSE681: Software Modelling and Analysis
Project #2

Key/Value database Implementation
Version 1.0

Instructor:
Dr. Jim Fawcett

Author:
Rakesh Nallapeta Eshwaraiah
SUID: 724573312

The implementation of the key/value database is based on the operational concept document designed in project #1.

The description for the implementation is given below:

- Test Executive package is used to test all the requirements of this project
- Structure for a database element is defined and created using DBElement package.
- Display package is used to display all the outputs on the console from different functionalities in a proper format which is readable
- All the processing on the database and its elements like insert, delete, fetching all keys are implemented in DBEngine package
- The queries for the fetching contents from database are implemented as functions in QueryEngine
- DBfactory is implemented to create immutable database from the query results.
- Timer package is used to design a scheduler for persisting the database at regular intervals. Scheduler package is named as Timer package in this project implementation
- Editing of the database elements are implemented as functions in the Item Editor package
- PersistEngine package implement the persistence of database contents to an XML file and also it implements augmenting the persisted files along with any changes back to main database
- Package structure and database dependency are loaded from the XML file and displayed onto the console. This is also implemented in PersistEngine package
- Error handling/exception handling are also implemented in this project
- New version of the database is stored in an xml file every time persist functionality is called
- All the functionalities are implemented for database having integer type key and value which is defined by DBElement
- For one the of the query requirement where the keys with specified pattern has to be displayed, the database used has a key of type string type and the value of the DBElement structure

Comparison

Most of the concepts discussed in OCD like database engine, scheduler, persist engine, DBFactory etc. are implemented as designed. But, sharding while persisting the database and compound querying concepts are not implemented.

- All the database contents are saved to the local machine in an xml file instead of partitioning the database contents and saving as separate file
- Simple queries are used for any fetching/modifying of the database contents. Compound querying has not been used to query database

- Authentication was discussed in the OCD but there was no requirement for its implementation so this project doesn’t have authentication implementation
- Item Factory is implemented within the DBEngine package. There is not separate package for item factory

The original concept designed in OCD was practical. Almost all the functionalities are achieved except sharding and compound querying, which can be implemented for large sets of data. There were challenges while implementing this project like XML handling, Parsing XML file, storing as immutable database, maintaining code complexity and size, etc.

The things which I learned during implementation did not make my OCD less relevant. But, there are few functionalities which can be achieved in different ways like, XML can be created using XDocument class, XML LINQ, payload wrapper given by professor and queries can be written using LINQ, predicate query etc. These different approaches helped me understand various concepts and its different ways of implementation which can reduce the complexity of the code. The usage of generics in the implementation helped in avoiding duplication of code for multiple datatypes.

Conclusion

The OCD played a vital role for this project’s implementation. Most of the components are implemented as it was designed in OCD except for sharding and compound querying. The OCD was practical enough to design and implement the project according to the requirements.