**Epipog**

**Specification**

**Collection Class Family**

**Dec. 28, 2016**

# 1. Class

The collection object is the definition for specifying a named collection. A collection consists of a data store representation and optionally a schema and one or more indices.

## 1.1 Fields

The base class defines the following fields:

private String collectionName; // Name of the Collection

This field is the name of the collection.

private DataStore store; // Data Store for collection

This field is the data store representation for the collection.

private Schema schema; // Schema associated with the collection

This field, if non-null, is the schema associated with the collection.

privateArrayList<Index> indices; // Index(s) for collection

This field, if non-null, are the indices for the collection.

private Index autoIncr; // internal id - automatic increment

## 1.2 Methods

This class contains the following implemented methods:

***Constructor***

public Collection( String collectionName );

The constructor for instantiating a collection object. The parameter collectionName is the name of the collection.

***Accessors (Getter/Setter)***

public void Schema( Schema schema );  
public Schema Schema();

These methods set and get, respectively, the schema associated with the collection.

public void Store( DataStore store );

This method sets the data store for the collection. If the data store is already set, an exception is thrown.

public DataStore Store();

This method gets the data store gets the data store assigned to the collection.

public void IndexAuto( indexAuto ) throws Collection Exception;  
public Index IndexAuto();

These methods set and get the automatic incremented internal index for the collection.

public void Parser( Parse parse );

This method sets the parser for parsing input from a file.

public Parse Parser();

This method gets the parser for parsing input from a file.

public void DeleteCollection();

This method deletes a collection from storage. If an error occurs, a CollectionException is thrown.

***I/O***

public void Open() throws CollectionException;

This method opens the underlying storage bound to the collection. If an error occurs, a CollectionException is thrown.

public void Close() throws CollectionException;

This method closes the underlying storage bound to the collection. If an error occurs, a CollectionException is thrown.

***Query***

public void InsertC( ArrayList<String> values ) throws CollectionException

This method inserts a row of data into the data store, where the data is aligned with the columns in the schema assigned to the data store. If an error occurs, a CollectionException is thrown.

public void Insert( ArrayList<String,String> values ) throws CollectionException

This method inserts a row of data into the data store, where each item in values is a key-value pair; whereby, the key is the field (column) name and the value is the value of the field. If an error occurs, a CollectionException is thrown.

public void Parse() throws CollectionException

This method parses an input file and inserts corresponding data as rows into the data store. If an error occurs a CollectionException is thrown.

public ArrayList<Data[]> Select( ArrayList<String> fields ) throws CollectionException

This method selects rows of data from a datastore. TODO