General Overview of the system, small user guide:

The system is broken into 3 phases. In the first phase, the user may give an input of a text file containing XML records, which will then be parsed and broken into 4 text files containing the parsed data. After that, Phase 2 will either load the index files from the newly made text files, or dump previously created index files for debugging purposes. After that, the user can call phase 3, which will organize the data into a database, and allow them to query it in a sophisticated manner. The user can specify a price range, a date range, a location, and a category, as well as multiple terms(or parts of a term if the user states the term is a wildcard). After the query is entered, Phase 3 will search the database in the most efficient manner it can and find the results, if any, that match the users query and return them.

To initiate phase one, run PhaseOne.py with python3, and then enter a valid text file, then it will parse it. For Phase two, run PhaseTwo.py with Python 3, and enter "L" to load the index files. For Phase three, just run PhaseThree.py with Python 3 and query the database as many times as required.

Detailed design of software, focus on components required to deliver the major functions:

Analysis-

In order to get the most efficient runtime, we take the keywords, and use the data intersect method to evaluate the common values. So, for example- if we have multiple queries, and we have a null set right after the first two queries, we would not need to go through the next queries.

We also see if, example- the location query comes with prices or dates, so that we can directly use the same database to get the values needed.

Testing strategy:

- Query language testing
  - Testing all test cases given in assignment specs and made sure the correct keywords/terms came up
  - Testing many query examples and added random white spaces in all locations
  - Testing multiple terms with/without wildcard placed in random parts of the query
- Data Parsing:
  - Tested on multiple text file sizes, and manually checked the processed data on the smaller files to match how it was supposed to look
- Queries
  - Compared query results to the text files we generate before the indexes, making sure all items that should return the right ID return.

Group work break down:

Thomas

- Price, Location, Date, Term and Category queries.

- Set intersection for combining queries like an AND.
- Making the files that were sorted in a format that bsddb3 would like.

Chady

- Complete Phase one, data parsing and organizing into text files
- Phase two, organizing data into appropriate idx files and allowing user to dump the data for debug
- Developed the query processing code for phase three, takes in a query and splits it into keywords with their operators and values, as well as a list of terms, stating if they are wildcard or not

Farish

- Check valid date
- Iterate through the keywords and choose the most appropriate database to use along with the required parameters and see the most efficient way to run the query