INTERMEDIATE MILSTONE

Blackbox tests

boolean addPublication (String identifier, Map<String, String> publicationInformation)

1. identifier and publication information is not null, returns false.
2. identifier is an empty string and the publication information is not null, returns false.
3. identifier is not null and the publication information is null, returns false.
4. identifier is not null and the publication information is an empty map, returns false.
5. identifier is not null and the publication information map is missing one or more required keys, returns false.
6. identifier is not null and the publication information map has all required keys and values, returns true.
7. Identifier is already present in the database, returns false.
8. publicationInformation values are duplicate, returns false.

Data flow:

1. calling this method at first
2. calling this method after addReferences
3. calling this method after addVenue

boolean addReferences (String identifier, Set<String > references)

1. identifier is null and references set is non-null, returns false.
2. identifier is empty and references set is non-null, returns false.
3. identifier has a valid value and references set is null, returns false.
4. Identifier has a valid value and an empty references set, returns true
5. Identifier has a valid value and a non-empty references set, with invalid values, returns false.
6. Identifier is non-null and a non-empty references set, with all valid values, returns true.
7. Adding references multiple times for the same publication, returns true

Data flow:

1. calling this method before addPublication
2. calling this method after addPublication
3. calling this method after addVenue
4. calling this method before addVenue

boolean addVenue (String venueName, Map<String, String> venueInformation, Set<String> research areas)

1. venueName is null or empty, returns false
2. venueInformation is null or empty, returns false
3. researchAreas is null or empty, returns false
4. venueInformation should have at least one valid key-value pair, returns true
5. venueInformation is non-null and non-empty strings, returns true
6. venueInformation has a key with an empty string value, returns false.
7. venueInformation has a null key or value, returns false.

8. venueInformation has characters that exceeds the maximum allowable length, returns false.

Data flow:

1. calling this method before addPublication
2. calling this method after add Publication
3. calling this method before addReferences
4. calling this method after addReferences

boolean addPublisher (String identifier, Map<String, String> publisherInformation)

1. identifier is null or empty, returns false
2. publisherInformation is null or empty, returns false
3. publisherInformation has atleast one key-value pair, returns true
4. publisherInformation is non-null and non-empty strings, returns true
5. identifier is an empty string, return false.
6. publisherInformation has a key with an empty string value, returns false.
7. publisherInformation has a null key or value, returns false.

Data flow:

1. calling this method before addPublication
2. calling this method after addPublication
3. calling this method before addReferences
4. calling this method after addReferences
5. calling this method before addVenue
6. calling this method after addVenue

boolean addArea (String researchArea, Set<String> parentArea)

1. researchArea is an empty string, return false.
2. If parentArea contains a null or empty string value, return true(can assume it is at the top of the research area hierarchy)
3. researchArea is duplicated in the library, returns false.
4. parentArea value does not exist in the library, returns false.

Data flow:

1. calling this method before addPublication
2. calling this method after addPublication
3. calling this method before addReferences
4. calling this method after addReferences
5. calling this method before addVenue
6. calling this method after addVenue

Map<String, String> getPublications (String key)

1. key is null or empty, returns an empty map
2. key is an empty string, returns an empty map
3. key does not match any publication in the library, returns an empty map.
4. Key is valid, return an map with valid data.

Data flow:

1. calling this method before addPublication
2. calling this method after addPublication

int authorCitations (String author)

1. author is null or empty, returns 0.
2. Author is empty string, return 0.
3. author does not match any author in the library, return 0.
4. author has a match in the library, return the count.

Data flow:

1. calling this method before addPublisher
2. calling this method after addPublisher

Set<String> seminalPapers (String area, int paperCitation, int otherCitations)

1. area is null or empty, returns false
2. paperCitation should be a non-negative integer.
3. otherCitations should be a non-negative integer.
4. area is an empty string, return an empty set.
5. paperCitation is zero and otherCitations is zero, return an empty set.
6. paperCitation or otherCitations are negative integers, return an empty set.

Data flow:

1. calling this method before addPublication
2. calling this method after addPublication
3. calling this method before addPublisher
4. calling this method after addPublisher

Set<String> collaborators(String author, int distance)

1. author is null or empty, returns empty set
2. distance should be a non-negative integer.
3. author is an empty string, return an empty set.
4. distance is zero, returns a set containing only the author.
5. distance is negative, returns an empty set.

Data flow:

1. calling this method before addPublication
2. calling this method after addPublication
3. calling this method before addPublisher
4. calling this method after addPublisher

Set<String> authorResearchAreas (String author, int threshold)

1. author should is null or empty, returns empty set.
2. threshold is negative integer, returns empty set.
3. author is an empty string, returns an empty set.
4. threshold is zero, returns an empty set.
5. threshold is positive, returns set of research areas.

Data flow:

1. calling this method before addPublication
2. calling this method after addPublication
3. calling this method before addPublisher
4. calling this method after addPublisher
5. calling this method before addReferences
6. calling this method after add References

Code Design

Java Files

a) DataInsertion.java - This file contains the following methods

1. addPublication - This method takes the publication information with the identifier and stores the data into the database where each key is an attribute and its corresponding value will be stored in its column. So calling this method will valid data adds a row into the database.

2. addReferences - This method will do a union operation on the set of references and the set of already existing references. The modified set is then replaced in the database.

3. addVenue - This method adds the venue by creating a table in the database if the method is called with valid data.

4. addPublisher - This method adds the details of the publisher into the database

5. addArea - This method adds the research area into the database.

b) DataRetrieval.java - This file contains the following methods

1. getPublications – This method gets the information that the publication contains along with the publications it is referencing in it.

2. authorcitations – This method returns the number of times the author has been cited in other publications. This can be done by traversing all the map objects and checking the author name, if the name matches, then the count can be incremented which will be returned at last.

3. seminalPapers – This method traverses the publication objects and returns all the publications that satisfies the given number criteria passed as the parameter.

4. collaborators – This method returns the author names who are the distance specified in the parameter.

5. authorResearchAreas – returns the areas and all its parent areas that the author has published at least “threshold” papers.

c) Conversion.java

1. main – This method takes an input file and adds the IEEE references to the file.

Data Structures

1. I will be storing the research Areas in a map where the key will be a research area and its value will be all its parent area. This will be stored in the form

Map<String childArea, list [ParentAreas]>

2. I will be using Graph in the collaborators method where authors will be the vertices and the distance between them will be the weights of the graph.

3. I will be using ArrayLists to store the Parent Areas of a particular research area. Then this list may be converted to a set to eliminate the possibility of duplications which may be necessary at the time of implementation.

Key Algorithms

Breadth first search – It is used to find the distance between the authors to know the potential collaboration within them. Djikstra’s algorithm can also be used to find the distance between the authors

Linear Search – It is used to traverse the Map objects to get the required data wherever neccessary.