Software requirements:

- **Java**
  This requires Java 8 to build the source.
  You may use Oracle jdk or openjdk
  On Ubuntu:
  > sudo apt-get install openjdk-8-jdk
- **Git**
  Download and install git on your system to pull the code from the repo.
  On Ubuntu:
  > sudo apt-get install git
- **Maven**
  Setup:
  1. Download maven at some location.
  2. Set the path for MAVEN_HOME and MAVEN_PATH to maven and maven bin OR add maven bin to the PATH

  On Ubuntu:
  > sudo apt-get install maven
- **Make**
  Download and install make to drive the build process
  On Ubuntu:
  > sudo apt-get install make
- **ElasticSearch**
  Setting up ElasticSearch:
  1. Download ElasticSearch 5.2.2 from following link
     ElasticSearch
  2. How to run ElasticSearch server on your system:

  a) For Windows XP/7/8/10 operating system
  Open command line and switch to directory where you have copied ElasticSearch folder.
  Type following commands in command line
  a) $ cd bin
  b) $ elasticsearch.bat

  b) For Mac operating system
  Open command line and switch to directory where you have copied ElasticSearch folder.
  Type following commands in command line
  c) $ cd bin
  d) $ ./elasticsearch

  c) For Linux (ubuntu)
  > wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-5.2.2.deb
  > sudo dpkg -i elasticsearch-5.2.2.deb
  > sudo service elasticsearch start

System requirements:

ElasticSearch server:

1. Java 8
2. 6 GB Heap allocation to Elasticsearch
   * Go to jvm.options file in elasticsearch home where config files are kept.
     > sudo vi /etc/elasticsearch/jvm.options
   * Modify -Xmx2g param to -Xmx6g
   * Restart the server
     On Ubuntu:
     > sudo service elasticsearch restart

JVM to run the application:

1. Min heap size 3GB
   * Run the application at the jar location using
     > Java -Xms3g -jar ReviewSearch-jar-with-dependencies.jar

Database files:

1. Dblp database files are present at following link:
   DBLP Data

Build the application:

1. Clone the git repository using the given url.
   > git clone https://github.ccs.neu.edu/CS5500-Spring2017/team19.git

2. Go to the team19 folder and find the Makefile
   - Build the code, run the test and generate the jar
     > make build

     Note: Make sure you have elasticsearch running on your local in order to have the test cases succeed while using this option to build.

     This will skip the test cases and build a jar
     'ReviewSearch-jar-with-dependencies.jar' at the target location ./Phase3/target

   - Build the code and generate the jar
     > make noTest

     This will skip the test cases and build a jar at the target location ./Phase3/target

- Only run the test cases
  > make test

  Note: this should have elasticsearch running so as to have all the test cases succeed

3. Run the application by executing the jar with required dependencies
   > Java -Xms3g -jar ReviewSearch-jar-with-dependencies.jar

   Note: The jvm by default limits the number of entities that can be parsed at once. Hence the above command doesn't load the whole data file. However, it can be used for testing (takes 3 mins as compared to 21 mins for the whole data).

   For full load use
   > java -Xms3g -DentityExpansionLimit=0 -jar ReviewSearch-jar-with-dependencies.jar

   Note: Make sure you have elasticsearch running with the required heap size so as to have all the functionalities work as expected.

4. Login in the application as given below to access it.

   Note: In order to have the query features working, load the data as mentioned below as the first step.

Assumptions:

- The system has the required amount of RAM in order to run elasticsearch and the application
- The files that is loaded as dblp data is strictly named as 'dblp.xml'. Also, the location should have 'dblp.dtd' when the data is being loaded.
- The data file for the committees should be placed in a folder called 'committees' and that location has only the file that has to be loaded for committee

  Note: Test files for UI are also included in the test suite under test/ui. These files need to be uncommented in order to run these tests.

Reviewer Search:

How to login:

1. By default application has username : admin and password : admin. Using these credentials user can log in.
2. Additionally user can create new user using Register option

How to loading dblp.xml:

1. After logging in. Click on File menu option present in top left.
2. File->New->Database
3. Above action will option dialog to select xml file.
4. Select dblp.xml file.(Do ensure dblp.dtd file is present in same directory along with dblp.xml)

How to load committees files:

1. After logging in. Click on File menu option present in top left.
2. File->New->Committees
3. Above action will option dialog to select xml file.
4. Select committees directory which has text files for all committees.

How to search:

1. After loading dblp.xml and committees data successfully. User can add multiple queries to perform search.
2. To add new query, press + button on bottom right.
3. User can choose either AND or OR radio button to either enforce all constraints or any one.
4. Title, Author, Journal expects string value to be entered in textbox. Allowed operation for string fields are

a) **Has any**: This will return all entries in which field contains any of the word entered.
b) **Has phrase**: This will return all entries in which field contains exact entered phrase.
c) **Has prefix**: This will return all entries in which field contains entered phrase as a prefix in a term.

4. Year, Volume expects integer value to be entered in textbox. Allowed operations are:

a) **Equal**: Returns all results with exact match
b) **Not equal**: Returns all results which doesn't contain entered value
c) **Less than**: Returns all results which contains value in field less than entered value
d) **Greater than**: Returns all results which contains value in field greater than entered value
e) **Less than or equal to**: Returns all results which contains value in field less than or equal to entered value
f) **Greater than or equal to**: Returns all results which contains value in field greater than or equal to entered value
g) **Between including**: Returns all results which contains value in field greater than entered value in first textbox and less than second textbox while including boundary values
h) **Between excluding**: Returns all results which contains value in field greater than entered value in first textbox and less than second textbox while excluding boundary values.

How to see author data:

1. Once search results are loaded after pressing search.

2. User can double click on any row, which will open respective author's profile dialog.
3. This dialog has two buttons

a) **Show Articles**: Clicking on this button will show list of all articles written by this author present in database.

b) **Show Committees**: Clicking on this button will show list of all committees of which selected author was member of.