Ks2
Magonzal
Mehta1

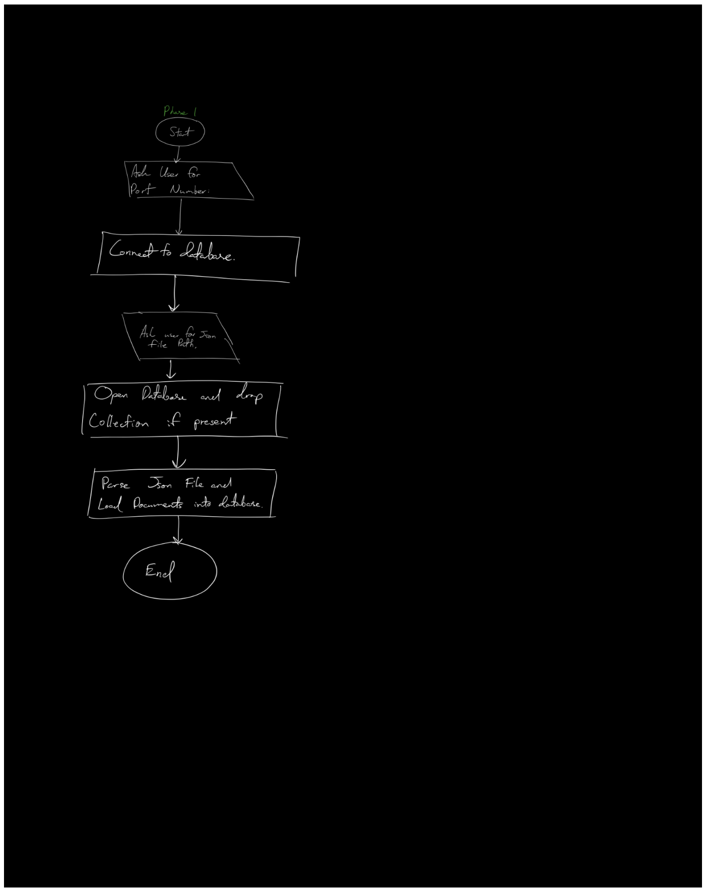
291 Group Assignment 2

General Overview and User guide

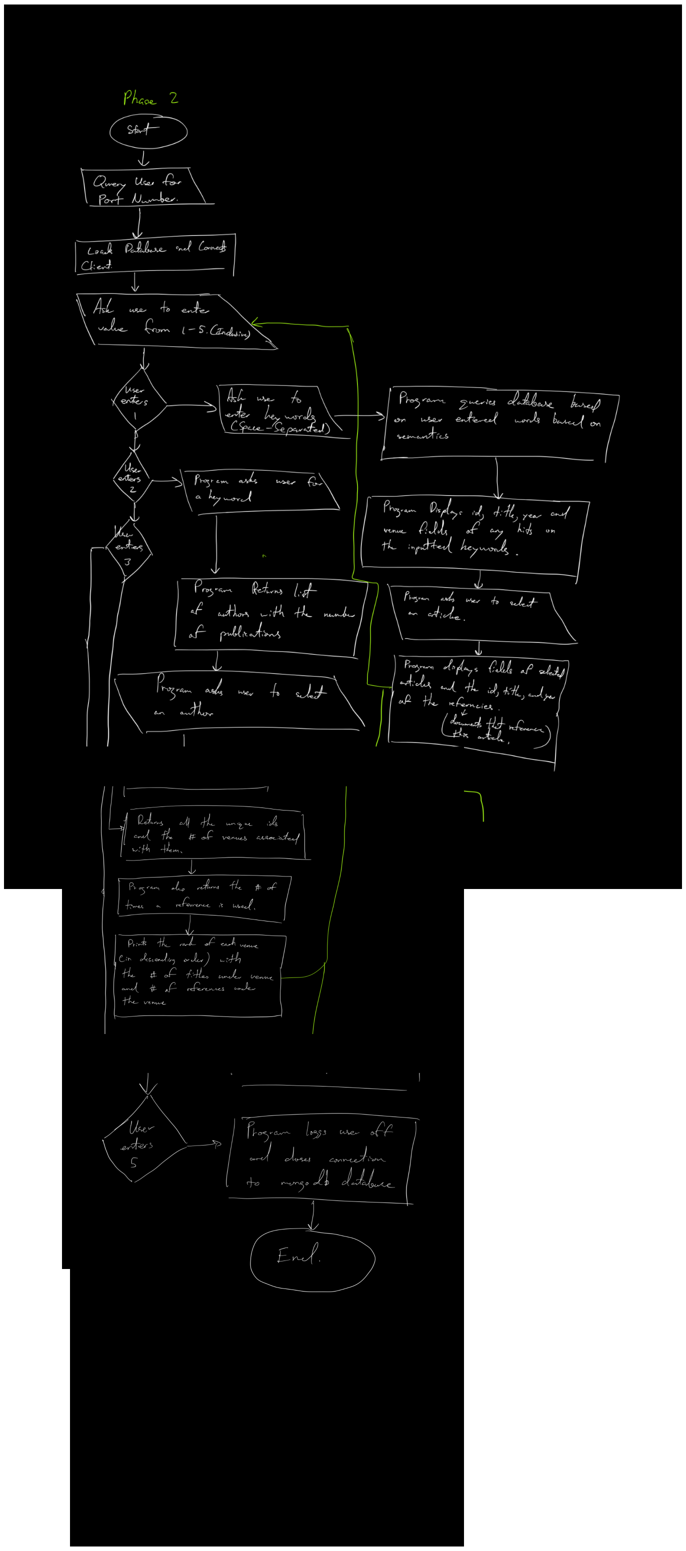
The designed system has functionality in two distinct phases. load-json.py populates the MongoDB database. Phase 2 allows the user to perform operations on the database created within phase 1. A user is able to search for articles, search for authors, rank venues by number of references to the venue, and add an article to the database.

A user will first run phase 1 with input of a port which Mongod is running on as well as a json file to populate the database with. Phase 1 can be run independently of phase 2. Once phase 1 is complete, phase 2 may begin with user operations. The user is prompted with 5 selections for the 4 operations the system can perform.

See the below diagram for a detailed view of program flow.


Phase 1
Start
Ask User for Port Number:
Connect to database.
Ask user for Json file Path.
Open Database and drop Collection if present
Parse Json File and Load Documents into database.
End

Ks2
Magonzal
Mehta1


Phase 2
Start
Query User for Port Number.
Load Database and Connect Client.
Ask user to enter value from 1-5. (Inclusive)
User enters 1
Ask user to enter key words (Space-Separated)
Program queries database based on user entered words based on semantics
Program Displays id, title, year and venue fields of any hits on the inputted keywords.
Program asks user to select an article.
Program displays fields of selected articles and the id, title, and year of the referencies.
(documents that reference this article.)
User enters 2
Program asks user for a keyword
Program Returns list of authors with the number of publications
Program asks user to select an author
User enters 3
Returns all the unique ids and the # of venues associated with them.
Program also returns the # of times a referrence is used.
Prints the rank of each venue (in descending order) with the # of titles under venue and # of references under the venue
User enters 5
Program loggs user off and closes connection to mongodb database
End.

Design of Software

There are two main classes within our software; Phase1 and Phase2. These classes exist independently of each other. load-json is responsible for the handling of the phase 1 of the project. A singular function exists to take user input as well as populate the database. The phase 2 object consists of 5 distinct methods; handle_1(), handle_2(), handle_3(), handle_4, and run(). handle_1() contains all functionality for the search of articles based on keywords in article titles. handle_2() contains all functionality for the search of articles based on keywords in author names. handle_3() contains all functionality for ranking of venues based on the number of times an article published in that venue is referenced. handle_4() contains all functionality for the insertion of a new article with abstract and venue set empty. run() provides a basic while loop for the cli to run through.

Testing Strategy

Our testing strategy was centered around building testing methods and operations as well as using the MongoDB VSCode extension to inspect that the database contains the expected elements. For example, we wrote temporary debug methods in each main such that we could branch coverage of our methods. Due to the nature of the assignment, we were able to separate the testing of phase 1 and 2. Likewise, each operation was tested independently with print statements along the way to ensure coverage and consistency. When operations did not function as intended, we first checked that the MongoDB operations were returning what was expected before beginning the debugging of Python code. The majority of bugs were errors in the initial setup of the aggregation pipelines and find operations.

Group Work Breakdown

Our group work breakdown involved splitting tasks along complexity. We determined that phase 1 was equivalent to a singular phase 2 operation. Thus, we delegated each individual a phase 2 task from requirements 1, 2, and 3. We collectively worked on the CLI, phase 1, and operation 4 together. Each group member put similar effort and time. Each group member spent approximately 5 hours on their work for the assignment. Coordination was primarily done through a project group chat as well as in person.

Assumptions

We assumed that phase 1 and phase 2 need to be completely independent programs. Likewise, we assumed a blank insertion meant inserting an empty string. We interpreted an empty string in a field to mean no value. Our insertion inserts blanks as “” and our solution for phase 2 requirement 3 disregards “” as a field for ranking.