(a) a general overview of your system with a small user guide,

The general overview of the system gives a high-level introduction and may include a diagram showing the flow of data between different components; this can be useful for both users and developers of your application.

The two programs combined allow user to load a collection from a json file and perform operations on the loaded collection.

The first program “load-json.py”, requires two arguments to run: the input file name and the connected MongoDB port number. The input file needs to be in the same directory.

Once the collection is load, the second program provides a command line UI for user to perform 4 operations on the loaded collection. User will be prompt to select the given menu by entering the corresponding index in the command line.

(b)

There are two programs. The first one loads a collection from a json file, and creates materialized views based on that collection. The second program is a command line UI providing four operations one the loaded collection.

In the second program, we have 5 main modules: Search, List the venues, Add an article, and dataHandler.

Search provide two search functions: search authors or search article.

List the venues module display top N referenced venues based on user input.

Add an article module gives user the option to add a new article in to the collection.

dataHandler contain all the mogoDB queries used in all other models.

Main is the part where we connect all modules together.

Here is our designed pseudo code for main.py:

```
While true:
   if (searchArticles):
       searchArticles()
   elif (searchAuthors):
       searchAuthors()
   elif (listVenues):
       listVenues()
   elif (addAnArticle):
       addAnArticle()
```

(c)

We used failure testing and error guessing strategy and try to achieve branch and statement coverage while testing. Unit testing was performed throughout the development process. We had 4 end-to-end use cases for testing.

Scenario 1: load file then perform article search.

Scenario 2: load file then perform author search.

Scenario 3: load file then list top venues.

Scenario 4: load file then add a new article.

In each scenario, failure testing and error guessing is performed by entering invalid inputs and testing edge cases.

Test cases were able to cover all statements and branches. We discover 3 bugs during final testing. The nature of all the bugs were the lack of input constraints on the user.

(d) your group work break-down strategy.

Split of tasks:

Tasks	**Assignee**	**Progress**	**Time Spent**
Search	Qi Zhou	Done	12hrs
List Venues	Tianyuan Fang	Done	12hrs
Add Article	Tianyuan Fang	Done	2hr
System design	All group members	Done	1.3hrs
Documentation	All group members	Done	30min
Database commands	All group members	Done	2hrs

Both of coordination:

Use discord to communicate and hold design meetings. Push code to GitHub for version control.