**Server Specification**

You are required to develop an information server for storing movie information. Your server will use the MySQL database as the back end storage and will allow users to query the database from a range of clients. Your server will need to provide the following functionality:

* Search: - Allows clients to search for a movie based on either title or director.
* Add: - will allow clients to add new movies into the database.
* Remove: - Will allow clients to remove data from a database.
* Update: - Will allow users to update certain information about a movie.
* Recommend: - Will recommend movies to a user based on their preferences
* Watch: - records what movies have been watched by a client.

In order to prepare for the implementation of the client you will need to design and implement a protocol that will allow the clients to communicate with your server. This may be done as a series of text statements. For example, if a user wants to search for movies the client could send the following string:

**Search My Left Foot**

The server should be able to process this statement and return a list of movies relating to the search term provided, My Left Foot. You should ensure that the server can handle any data sent to it by the client i.e. multi-word search terms.

All data must be returned to the client in JSON format. For this, you will need to create a JSON object to be returned to the client with all of the relevant results.

When a search is performed, results should be cached in a map data structure so that if the same search is repeated, possibly by another user, the system does not need to call the database again.

**Suggested Approach**

**Design Protocol:**

The first thing you should do is design your protocol. You will need to think carefully about what parameters are need for each function and how the server will interpret this request from the client. When a line is received by the server it will need to be able to identify which function the client wants to perform and then to read all of the relevant parameters needed to perform that function.

**Design JSON format**

The next step should be to design what structure you will use to return data to the client. As each movie will have multiple pieces of information the client will need to know what information is related to what field. JSON can do this easily, however you will need to design an appropriate structure for the JSON format. Also consider how you will represent multiple movies.

**Design your Data Access Object**

You should develop one class for communicating with your database. This class should contain all methods that require database interaction. For example, you may need to have a method *getMovieById*() which will search for a movie by ID.

**Identify other classes:**

Once you have designed your DAO you will then need to determine what other classes you need for your application. For example, you will need an interface class i.e. the class that handles the clients’ requests.

**Build your classes**

Once you have designed you can proceed to building them. You can develop a simple client, which will read commands from the command line and pass them to the server to simulate a client passing data in. You may then print the server’s response to the command line to see if the data returned is correct.