trec_eval can be considered an external service as we are invoking code that is external to our application.

Systematic Review Application

Description: The purpose of this application is to enable researchers to perform a systematic review using PubMed. A systematic review consists of a number of stages, which need to be recorded: (1) query construction, (2) Abstract Pool Evaluation and (3) Document Pool Evaluation.
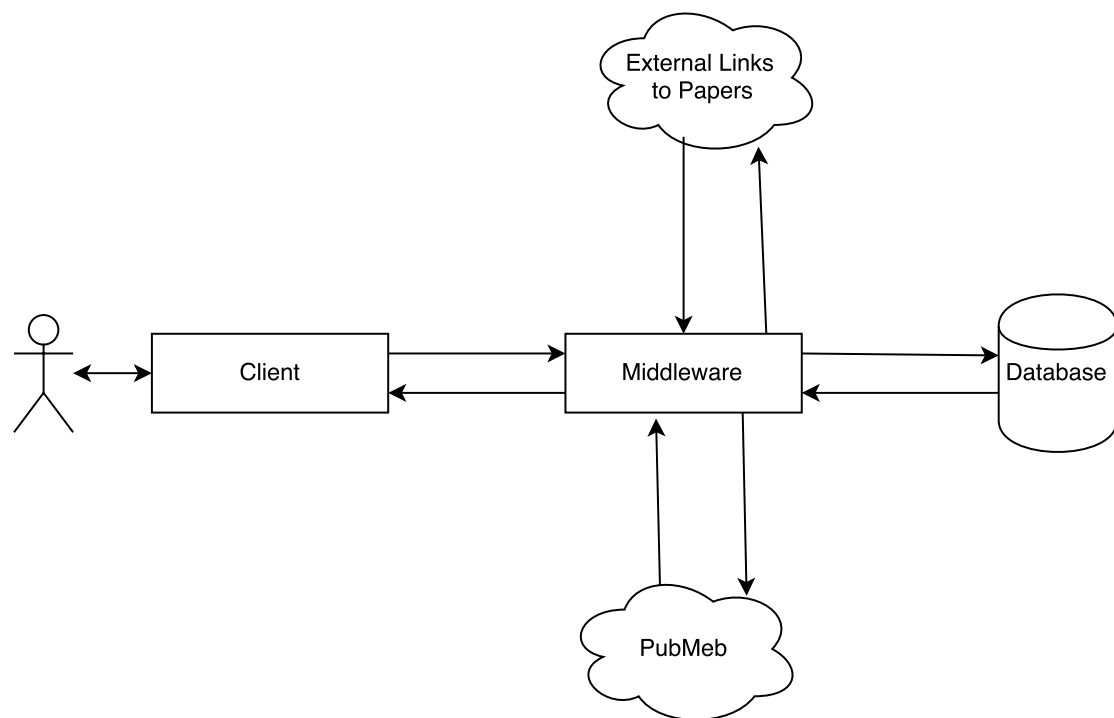
Specifications:
Researchers need to be able to:
- create an account and profile
- edit, update and maintain their profile
- create/edit/delete systematic review topics
- construct boolean based queries
  - for experts (a standard query box)
  - for novices (a structured / visual querying interface)
- view the total number of hits that a query would return
  - ~~(and how much overlap with the abstract pool)~~
- save/edit/delete queries
  - saved query results are added to the abstract pool
- add/remove results from a query to/from the abstract pool
- update the abstract pool based on the set of queries for that systematic review
  - this needs to check if documents that have been assessed will be removed, and warn the user.
- View the abstract pool of documents
- Judge abstracts in the abstract pool
  - If considered relevant, the document is added to the document pool
- View the document pool
- View the document in the document pool (perhaps through an external link)

- Judge the document in the document pool and mark as relevant or not-relevant
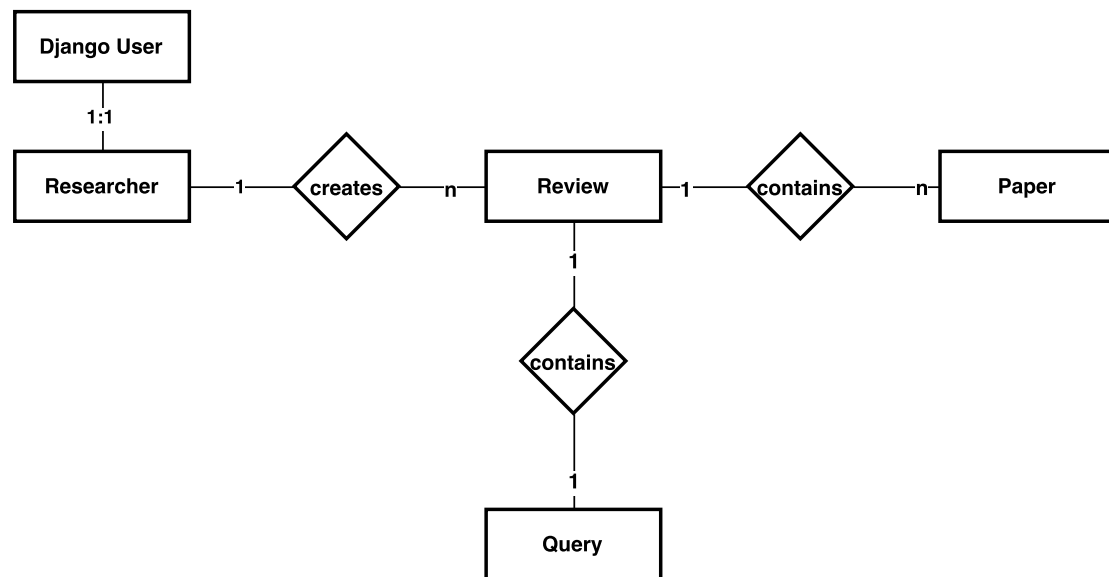- View final set of relevant documents

One researcher can create many systematic reviews, one systematic review contains a number of queries, each query retrieves a number of documents, which are added to the abstract pool. An abstract can be judged relevant and is added to the document pool. A document in the document pool can be judged relevant and added to the final pool.

The API to use is provided by E-Utilities which gives access to the PubMed api: http://www.ncbi.nlm.nih.gov/home/api.shtml

PubMeb is an external API (must have), and the External Links (may or may not be necessary) denotes that we could be contacting other sites for information about the a paper).

Actually the external links cloud should connect to the client – not the middle ware.

Researcher (user)

username, password, email stored in django user table.. so research is not needed

Review (user, title, description, date_started, query_string, pool_size, abstracts_judged, document_judged)
Paper(review, title, authors, abstract, publish_date, paper_url, abstract_relevance (T/F), document_relevance (T/F), notes)

A **Query** class could also be created. Here I have it as a one to one, so I subsume it within the **Review** class, but it could be represented as a one to many, i.e. a review has many queries (or many query parts) that are issued.

In Review, I have included a couple of attributes regarding the size and number of pool and abstracts/documents judged. We could compute them each time, but this would require numerous calls to the database.

Example Queries:

Attention Deficit Disorder

```
1 Attention Deficit Disorder with Hyperactivity/
2 adhd
3 addh
4 adhs
5 hyperactiv$
6 hyperkin$
7 attention deficit$
8 brain dysfunction
9 or/1-8
10 Child/
```

```
11 Adolescent/
12 child$ or boy$ or girl$ or schoolchild$ or adolescen$ or
teen$ or "young person$" or "young people$" or youth$
13 or/10-12
14 acupuncture therapy/or acupuncture, ear/or
electroacupuncture/
15 accupunct$
16 or/14-15
17 9 and 13 and 16
```

Back Pain

1. randomized controlled trial
2. controlled clinical trial
3. randomized
4. placebo
5. drug therapy
6. randomly
7. trial
8. groups
9. or/1-8
10. (animals not (humans and animals))
11. 9 not 10
12. dorsalgia
13. exp Back Pain/
14. backache
15. (lumbar adj pain)
16. coccyx
17. coccydynia
18. sciatica
19. sciatic neuropathy/
20. spondylosis
21. lumbago
22. exp low back pain/
23. spondylosis.mp. or exp Spondylosis/
24. back pain.mp.
25. or/12-24
26. 11 and 25
27. Ultrasonics/
28. ultrasonic*
29. ultrasound
30. or/27-29
31. 26 and 30

Federated Health Search Application