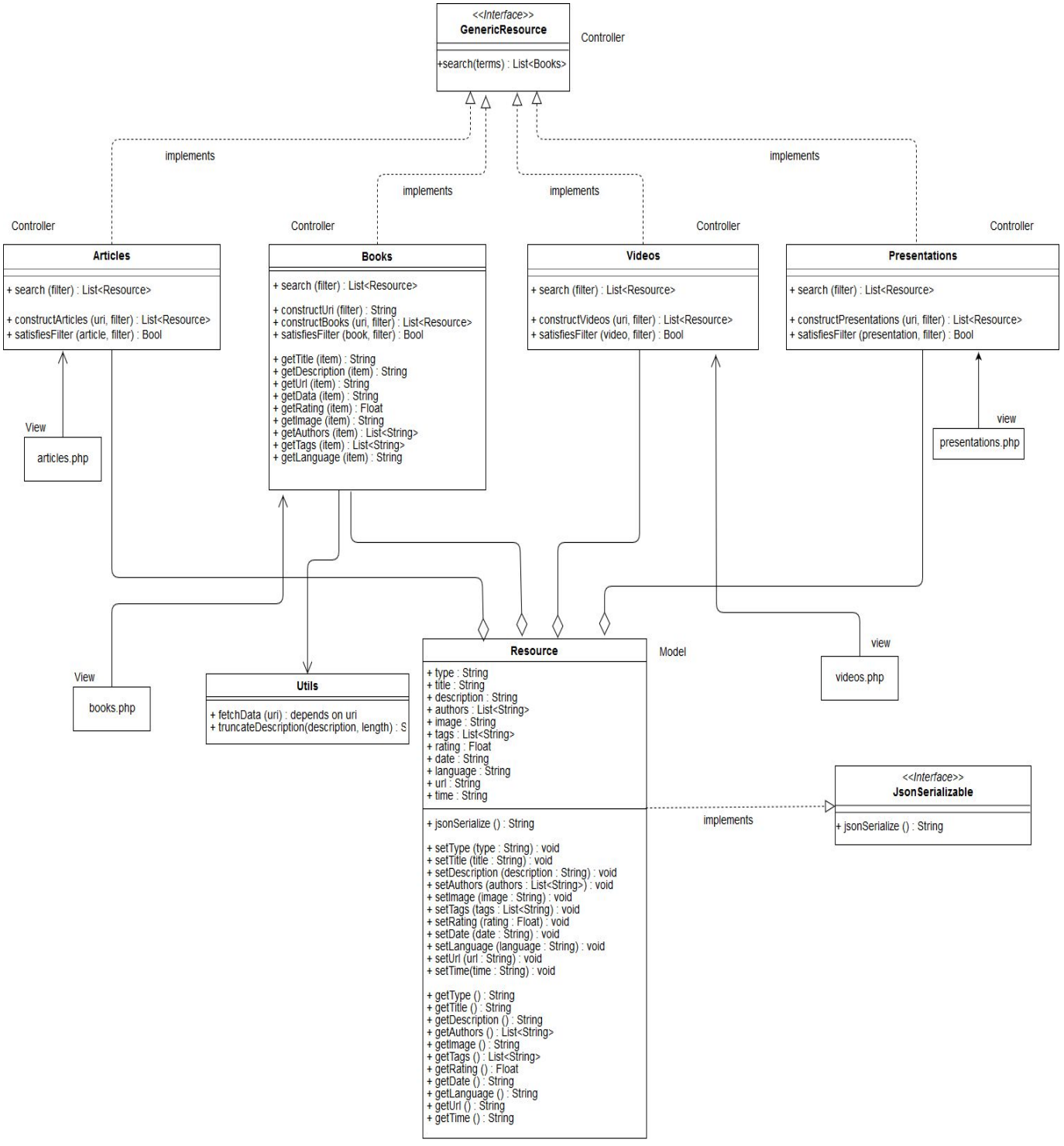
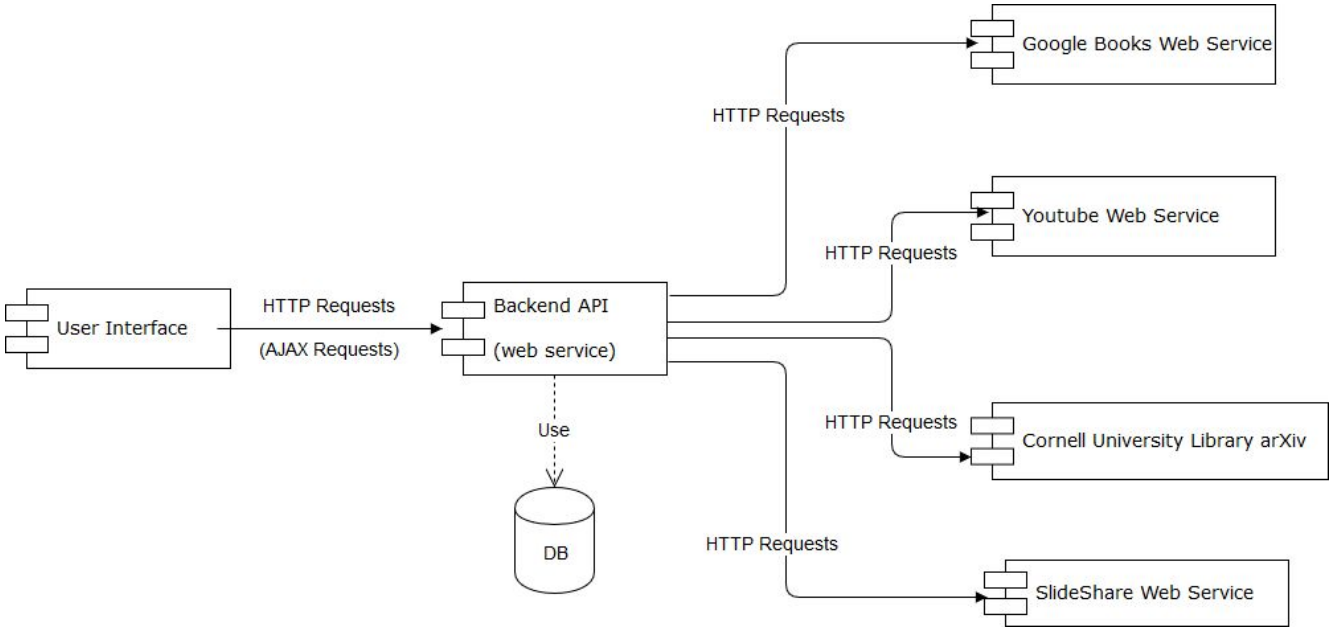
TReX (Topic-based Resource eXplorer) **Architecture**

1. UML Diagrams

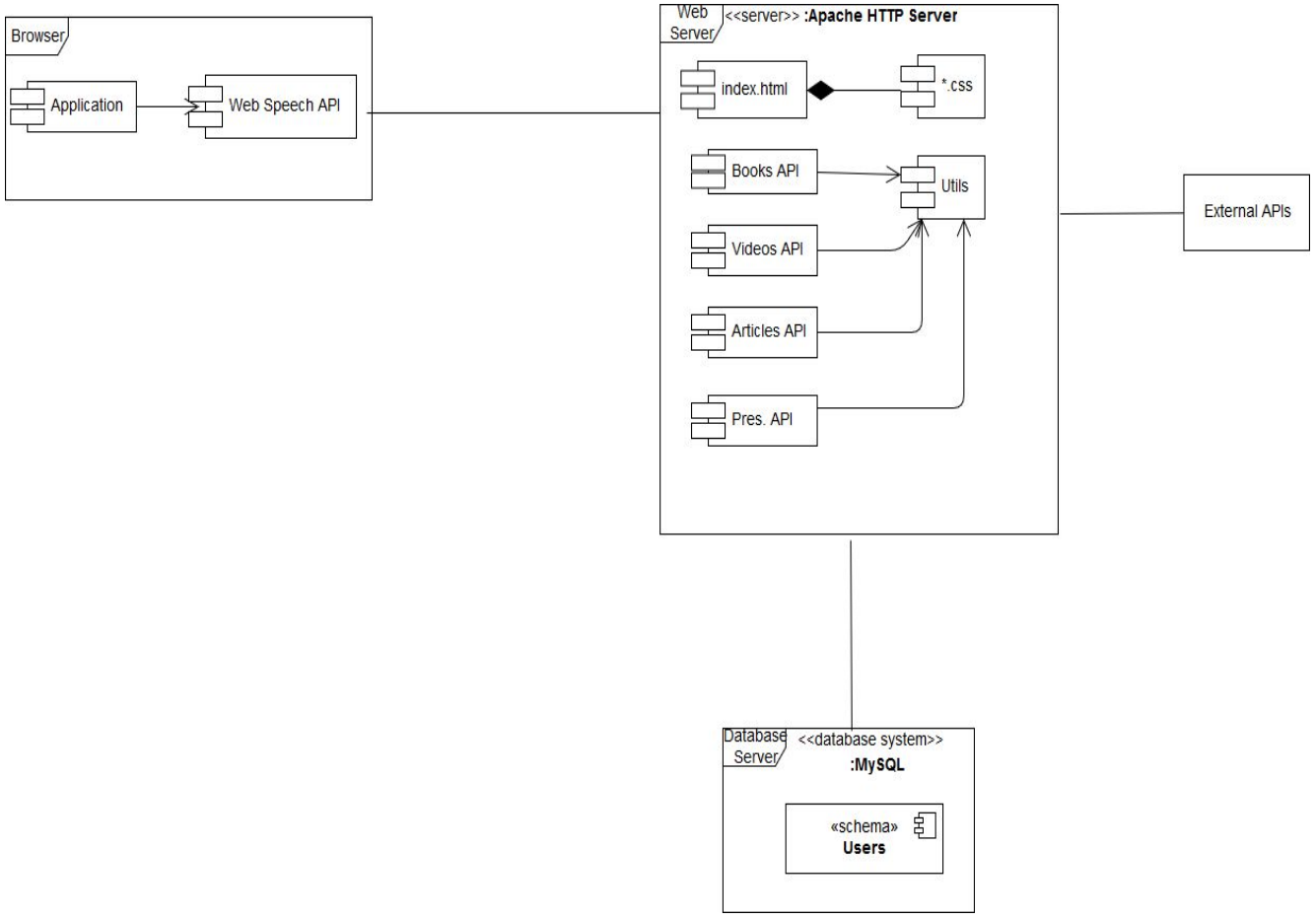
1.1. Class Diagram



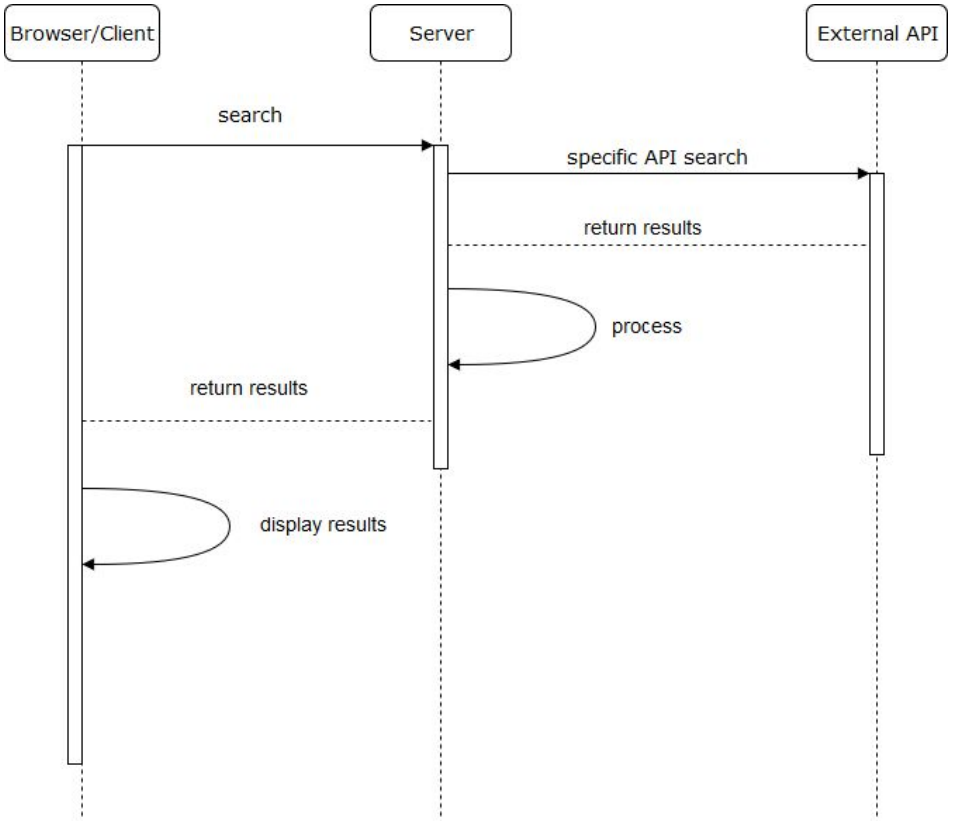
1.2. Components



1.3. Deployment



1.4. Sequence



1.5. Database Schema

We have opted for a relational database because:

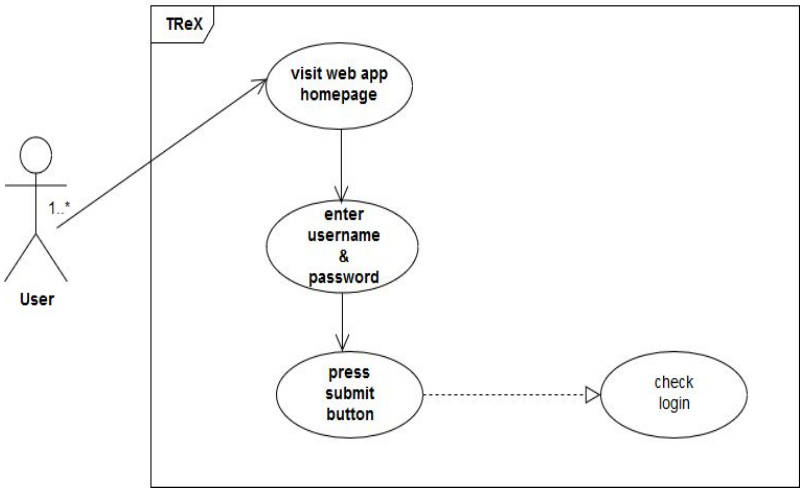
- our schema is simple
- the number of requests will be relatively small and a SQL database will scale
- we are already familiar with this technology

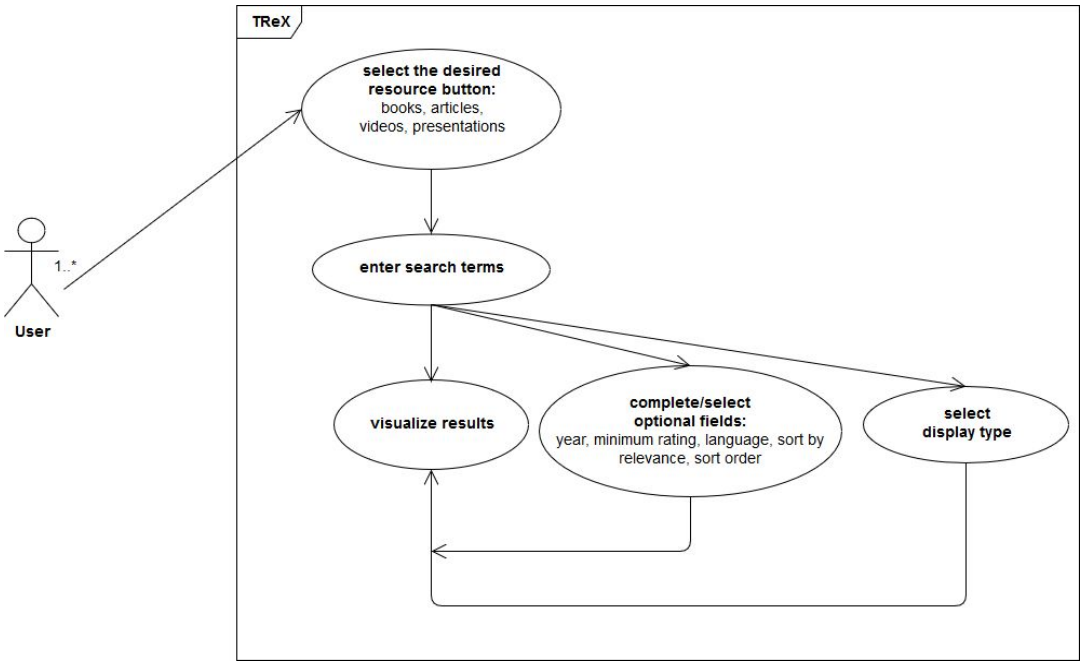
The first time when a user connects to our application, a hash function will be applied to the password entered by user and the result of the hash function will be stored in the database in the password field. Each time thereafter, when a user wants to connect, the result of the hash function applied to the password it is compared with the information stored in the password field next to the user with that username.
The email field will store the user's email address, being useful for password reset.

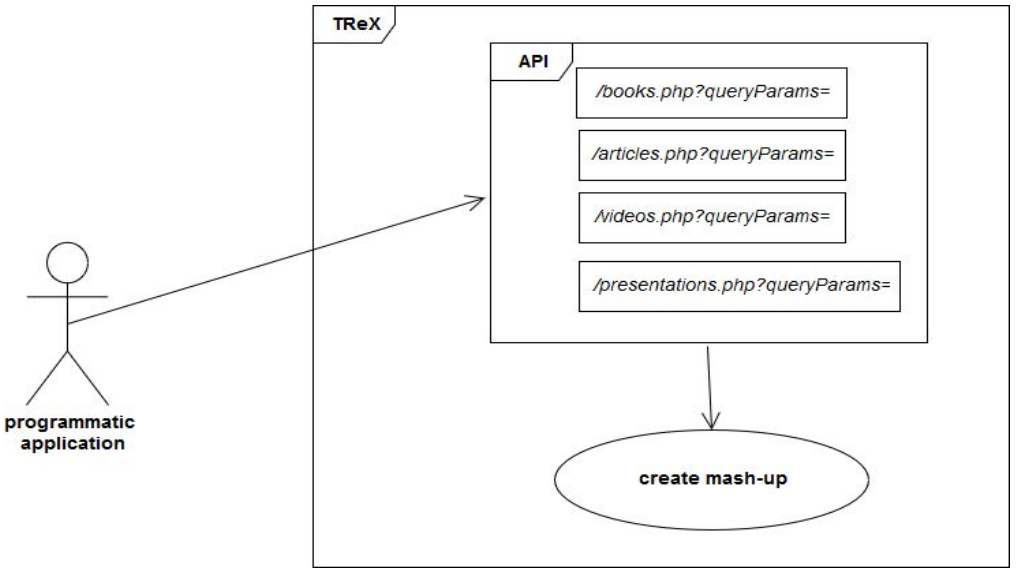
```
USERS
PK id NUMBER not null generated always as identity (start with 1, increment by 1),
   username VARCHAR2 (20) not null unique,
   password VARCHAR2  not null,
   e-mail VARCHAR2 (50)
```

2. Use Cases

Our application addresses both regular and programmatic clients.


TReX
visit web app homepage
enter username & password
press submit button
check login
1..*
User


TReX
select the desired resource button:
books, articles, videos, presentations
enter search terms
visualize results
complete/select optional fields:
year, minimum rating, language, sort by relevance, sort order
select display type
1..*
User


TReX
API
/books.php?queryParams=
/articles.php?queryParams=
/videos.php?queryParams=
/presentations.php?queryParams=
create mash-up
programmatic application