## 

## Database and Info Systems

Project Description

## CIS 550

Fall 2015

Professor Susan Davidson

**Group member**

Xiao Hu

Hiroyuki Kubota

Yi Xia

Wenhe Zhang

**Motivation**

The motivation of this project is, for a given database on movies, designing a website that can take user’s key words to provide comprehensive movie information corresponding to user’s inputs. Also as a movie recommendation tool, the designed website would suggest similar movies based on users’ search. Similarities of movies will be calculated based on data provided by the database.

To access this website, users need to connect their Facebook accounts (where it is assumed that all users own a Facebook account) with the website. Users can get knowledge on the movies they searched from our website. Additionally, users can make comments on the movie in website pages. Furthermore, they might find similar movies in this website.

**Features**

· Primary Features

Fig. 1 shows use cases in terms of this web application.

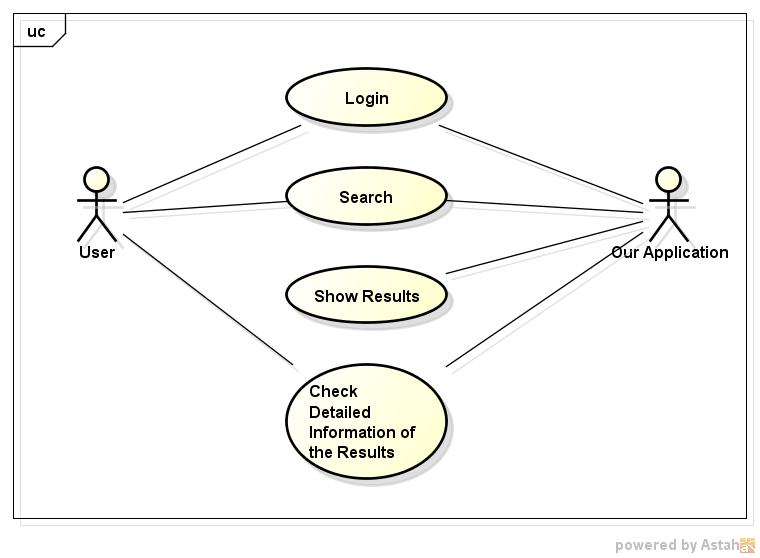


Fig. Use Case Diagram

The application contains four key use cases, “Login”, “Search”, “Show Results”, “Check Detailed Information”, and each case has the following feature:

- Login

Users should access the website by connecting their Facebook account.

* Search

Users can type in key words they want to search. Based on given input or inputs, the website will find several most similar results for users. To define similarity, distance will be used as a measure. The bigger the distance is, the less the similarity will be. The distance is defined as a Euclidean norm based on countries, genres, popularities, release dates and keywords. Those categorical data, such as keywords and genres, will be converted to numerical data. Then a numerical output will be calculated and to be used as distance.

* Show Results

After searching based on given inputs, the website will demonstrate the input information, as the center feature, in the center of the output device (the monitor, maybe), with related information from the database as branches shown around the center feature. Different pieces of movie information, grouped by movie names, will be located with different distances to the center features according to their similarities with the center feature.

- Check Detailed Information

When the users put the cursor on a movie, the website will show all other related information from the database associates with the movie.

· Additional Features

Some features might be implemented in the application, given enough time for the project.

* Key words detection

Users might make typos when input their key words. The website could automatically use some similar words as key words for further work.

* Comments

Users can add their own comments on the detail page of a movie.

Using NoSQL.

* Bing Search

Searching movie pictures and external information from Bing search engine.

**Technology and Tools**

BCE architecture will be implemented in this project. In each node, different technologies and tools will be used.

* Boundary - This represents GUI of the software. HTML and JavaScript will be used in this part. Specifically, Angular.js, Arbor.js and JQuery will be needed. The following screen shot shows an example of a search result:

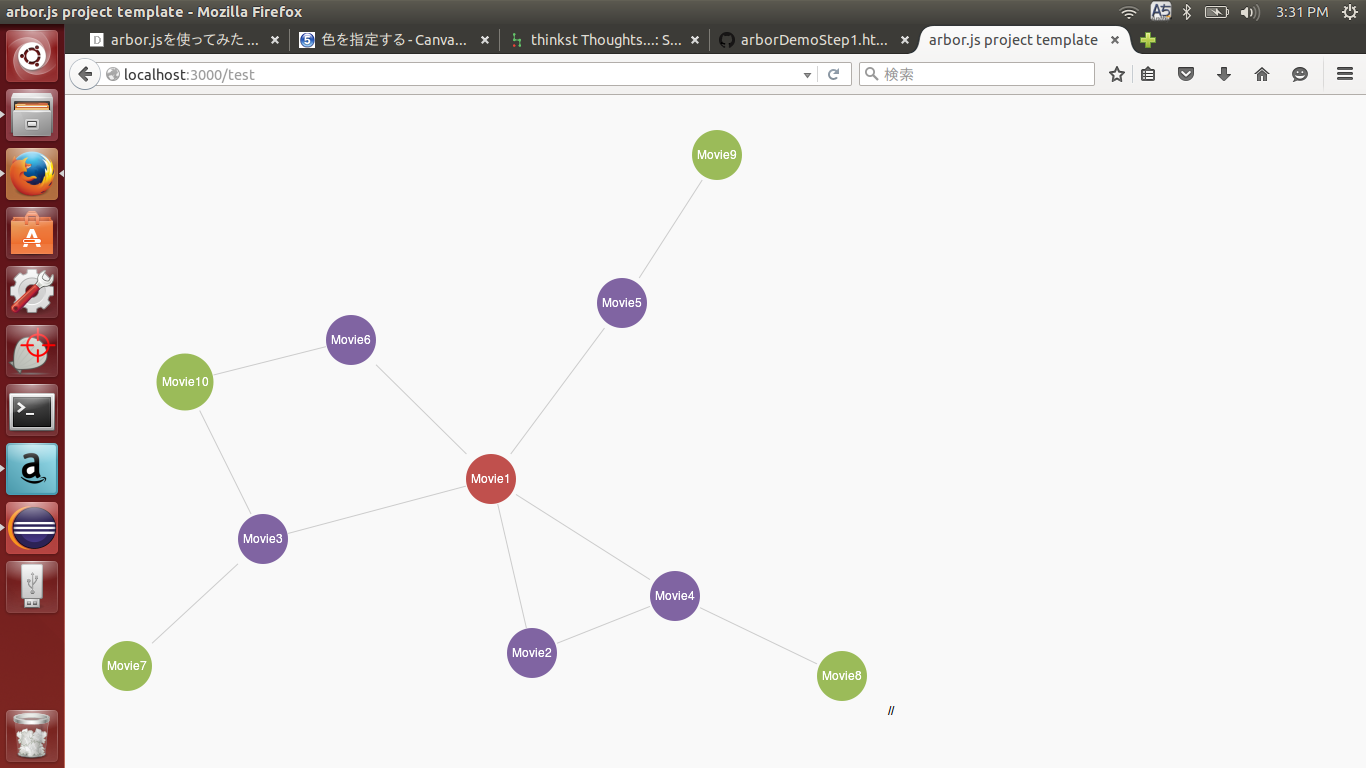


Fig. An Example of a Search Result

* Controller - Controller will manage data transformation between Boundary and Entity. Node.js and Express will be used for this purpose.
* Entity – MongoDB and MySQL will be used here. MongoDB will be basically used for storing user information, while MySQL will be used to store the given movie data. Syntax translations will be necessary in this part.

Fig. 3 and Fig. 4 summarize the technology and the tools.



Fig. 3 Physical Architecture



Fig. 4 Logical Architecture

**Schema Design**

MySql:

Movie (MId: int(11), title: varchar(255), popularity: double, rdate: date)

CountryLanguage (Mid: int(11), Country: varchar(255))

Has (Mid: int(11), Keyword: varchar(255))

Contain (Mid: int(11), Genre: varchar(255))

Details (Mid: int(11), rating: double, trailer: varchar(255), revenue: double, HP: varchar(255), budget: double, poster: varchar(255))

Compare (Mid: int(11), CMid: int(11), score: double)

Studio (Sid: int(11), name: varchar(255), HQ: varchar(255), logo: varchar(511))

Created\_by (Mid: int(11), Sid: int(11))

Child\_of (PSid: int(11), CSid: int(11))

Person (Pid: int(11), name: varchar(255), Dob: date, Profile: varchar(255))

Act (Pid: int(11), Mid: int(11), CharacterName: varchar(511), OrderNum: int(4))

Work\_on (Pid: int(11), Mid: int(11), department: varchar(255), job: varchar(255))

NoSql:

Comments{

‘Mid’:(Movie ID),

‘comment’:(User Comments)

}

**ER-Diagram**

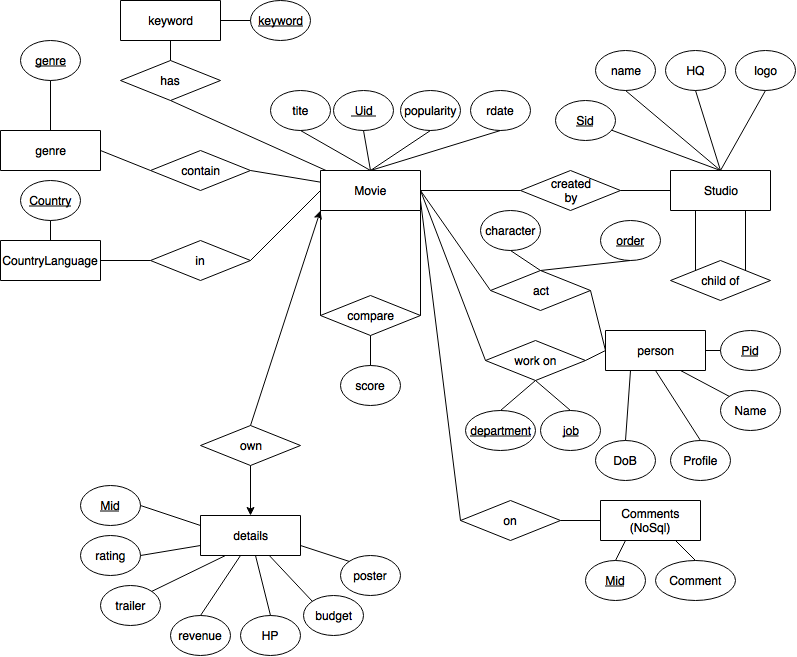


Fig. 5 ER Diagram