

CS 7460: Final Project: Collaborative Software Option

MovieNotes - Lokesh Balakrishnan

Introduction

The aim of this project is to build a collaborating tool which allows one to talk about Movies with their friends, maintain to watch lists based on this distributed knowledge. It would be something like a closed and trusted community like Facebook's secret groups which would allow the users to get some idea to decide on what to watch next during their free time from their friends' experience on watching different genres of movies.

Key Pains being solved, in this application we eliminate the need to maintain lists of movies one intends to watch during free time and manual maintenance cum sharing activities. In today's world, a movie buff would search for reviews on the internet, maintain to-watch lists as notes on notepad / ever note / to-do apps. Using the word of mouth reviews and interactions with pals on Facebook and tweets one would finally rate the list of movies. As soon as one completes watching the movies he would express reviews on social networking sites, in this way he communicates with his friends passing on the reviews. This project would integrate all these pieces implementing a distributed collaboration approach to aid users selecting their movies.

Motivation

My personal experiences and requests from friends to build an application, which helps one seamlessly decide/maintain/share movies list have together motivated me choose this project. The distributed cognition concepts learnt does have a vital role to play as well. As this application is just a To-Watch list for stand-alone users. The moment the user adds friends and the group collaborates, the knowledge from friends drives the user's decisions and the vice versa creating a win-win situation for the group. Since the group is closed we don't expect major trust issues.

Existing Solutions

Evernote plus Facebook, Most internet users utilize list maintenance tools like Evernote and social networking sites to obtain and share movie reviews. The combination of these two is widely popular and a lot of people use this option.

MovieLens is page which gives us movie recommendations based on our interests and the match is obtained from a domain of public users who share similar interests to us.

Movie Forums, a lot of movie forums are available on the web and these are like general discussion forums where the online community talks about various movies and anonymous users share movies list for various genres.

Proposed Solution and difference

Our application is different from the models is a couple of ways. Our application would be a closed group meaning the collaboration happens only between a trusted set of users. Movielens gives recommendations based on machine learning approaches; our software is plainly representing information from one's friends in a simple way with simple calculations and let the user decide what to do next. Most lists maintained on apps and notepads are not seamlessly shareable; our software is a shared component itself. The online reviewing and discussions form the first filter which makes the movie viewable but the decision towards arranging/prioritizing this set of viewable movies might be the problem solved by this software.

Application – MovieNotes

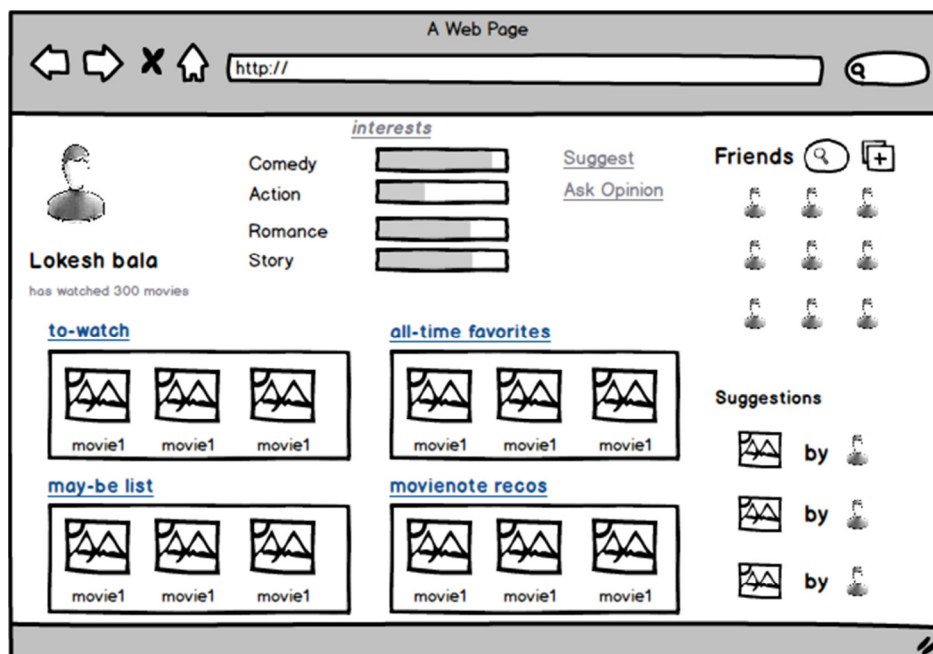
Our application would be Web Application where a movie buff would simply sign up, invite friends, form groups and collaborate to maintain their movie preferences and lists. Since IMDB API's aren't exposed completely we intend to use <http://developer.rottentomatoes.com/>

Features include

- Create/maintain to-watch lists
- View list of movies friends have watched
- Search movies (linked with rotten tomato apis)
- Rate/comment on movies
- Set interests and preferences
- View recommendations (system would indicate a system recommendation listing based on movie rating plus interests match percentage plus aggregate friends rating)
- Ask/provide Suggestions from/to specific users.

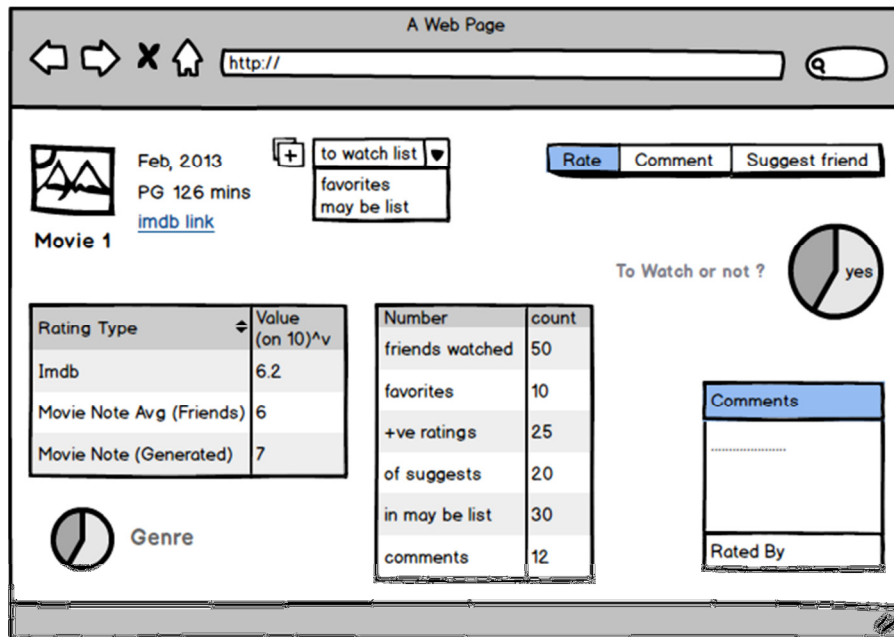
Screens

Profile page



[next Page ->](#)

Movies page



[<-prev page](#)

Plan – Timeline

Week 1	Setup frameworks (Google App Engine, Github) Look into Rotten Tomato apis & imdb apis
Week 2	Design UI screens for the front end Integrate the changes from feedback
Week 3	Design back end schema Write stubs for exposing RESTful apis
Week 4	Code front end screens and integrate with the template apis
Week 5	Code back end apis and Perform unit testing
Week 6-7	Integration, Testing & bug fixing + Prepare for presentation and delivery.

References

1. <http://movielens.org/html/tour/movies.html>
2. <http://developer.rottentomatoes.com/>
3. <https://evernote.com/>
4. Desjardins, Audrey, Carman Neustadedter, Saul Greenberg, and Ron Wakkary (2014). [“Collaboration Surrounding Beacon Use During Companion Avalanche Rescue.”](#) Forthcoming, *Proceedings of CSCW 2014*, Baltimore, MD, February 2014.