Brainstorming Specific Needs

No screens

Prof. Lydia Chilton
COMS 6998
21 September 2018
You already know front-end web dev:
HTML, JavaScript, Bootstrap, jQuery

And design:
Iterative design, critique

You will learn back-end web dev:
• Server-side programming (Flask),
• Databases (Sqlite, SQLAlchemy)
• Real-time Communication (Socket.IO)

And practice web design by:
• Rebuilding IMDB.com
• Rebuilding twitter
• Pursuing your own project
Last week’s high level goal:

• Remake the basic functionality of IMDB.com:
  • User goal: “Look up what actor was in what movie and then see all the other movies that actor is in.”

• Must store data in a **SQLite database**
• Must query it with **SQLAlchemy**
• Must Enable **CRUD** operations (create, read, update, delete)
  • This is not actually user-facing functionality of IMDB, but it’s essential back-end dev)
• Must use at least one **Database Join**
Dictionaries vs. Database Tables

movies = {
    '1': {
        'title': "Crazy Rich Asians",
        'rating': "PG-13",
    },
    '2': {
        'title': "Mission Impossible: Fallout",
        'rating': "G",
    }
}

actors = {
    '1': {
        'name': "Henry Golding",
        'year_of_birth': "1987",
    },
}

"Movie" table

<table>
<thead>
<tr>
<th>id</th>
<th>title</th>
<th>rating</th>
<th>year</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Crazy Rich Asians</td>
<td>PG-13</td>
<td>2018</td>
</tr>
<tr>
<td>2</td>
<td>MI: Fallout</td>
<td>G</td>
<td>2018</td>
</tr>
</tbody>
</table>

"Actor" table

<table>
<thead>
<tr>
<th>id</th>
<th>title</th>
<th>Year_of_birth</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Henry Golding</td>
<td>1987</td>
</tr>
<tr>
<td>2</td>
<td>Constance Wu</td>
<td>1982</td>
</tr>
</tbody>
</table>
Join tables

"Movie" table

<table>
<thead>
<tr>
<th>id</th>
<th>title</th>
<th>rating</th>
<th>year</th>
<th>Actors_backref</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Crazy Rich Asians</td>
<td>PG-13</td>
<td>2018</td>
<td>(automatic)</td>
</tr>
<tr>
<td>2</td>
<td>MI: Fallout</td>
<td>G</td>
<td>2018</td>
<td></td>
</tr>
</tbody>
</table>

"Actor" table

<table>
<thead>
<tr>
<th>id</th>
<th>title</th>
<th>Year_of_birth</th>
<th>Movie_backref</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Henry Golding</td>
<td>1987</td>
<td>(automatic)</td>
</tr>
<tr>
<td>2</td>
<td>Constance Wu</td>
<td>1982</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Tom Cruise</td>
<td>1962</td>
<td></td>
</tr>
</tbody>
</table>

Movie_actor_join

<table>
<thead>
<tr>
<th>Movie_id</th>
<th>Actor_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
</tr>
</tbody>
</table>

Actors.query.get(3).movie_backref
Movies.query.get(1).actors_backref
10-min Partners Discussions:

• Show your partner your interface and how a user navigates it.
• Share one thing you learned
• Show them your code for how you insert things in the join table.

• What other information would you like to include in the database?
  • How would you store it and query it?
Next week:

Build a web app that meets a specific user need in the movie data domain

(That is not a specific need that IMDB supports well)
Specific User Needs

Review!
General goals vs. Specific user needs

1. Help people find stuff
2. Help people find movies they like
3. Help people look up actors in films
4. Help people figure out who the leading actor in a movie.
What’s appealing about general goals?
General Vs. Specific Goals

Goal 1: “Clean the house.”

Goal 2: “Fold that basket of laundry.”

General goals sound appealing, but specific goals are actionable:
What **person** will take what **action** on what **object**?
Example: **Domain** and **Specific Need**

**Domain**: conference talks

**Specific need**: A timer that first counts down the talk time, then counts down the q&a time so the speak knows if they are on track.
### Domain

- **Amazon**: Online shopping
- **Facebook**: Social Network
- **Gmail**: Email client

### Specific Need

- **Uncommon books**
- **Harvard students to look up each other’s classes**
- **Web-based**
  - No page reload
  - Never Delete

### Generalized to

- **Clothes, Food, Amazon Fresh, Other sellers**
- **Ivy League, US Colleges, Everybody**
- **Chat, GDrive**
Every application starts with a specific need then generalizes.

**Domain:** conference talks

**Specific need:** A timer that first counts down the talk time and q&a

**Generalization:**

- Medium talks (20 min)
- Short talks (1 min) * 40 1-min intros in 1 hr
- Long talks (1 hr) * Might take questions during the talk.
Brainstorming Specific Needs for IMDB data
Coming up with the perfect idea can be intimidating.

The best way to have a good idea is to have lots of ideas.

- Linus Pauling
Only rule of brainstorming: No criticism
If you HATE someone’s idea, what should you do?

Have another idea.
“But IMDB does everything already.”
What are specific user needs of Craigslist that other companies serve better?
IMDB has lots of data just waiting to meet a specific user need

Storyline

The story follows Rachel Chu (Wu), an American-born Chinese economics professor, who travels to her boyfriend Nick's (Golding) hometown of Singapore for his best friend's wedding. Before long, his secret is out: Nick is from a family that is impossibly wealthy, he's perhaps the most eligible bachelor in Asia, and every single woman in his ultra-rarefied social class is incredibly jealous of Rachel and wants to bring her down. Written by JAP

Plot Summary | Plot Synopsis

Plot Keywords: nouveau riche | materialism | gold digging | aeroplane | chinaman

Taglines: The only thing crazier than love is family

Genres: Comedy | Romance

Motion Picture Rating (MPAA)

rated PG-13 for some suggestive content and language | See all certifications »

Parents Guide: View content advisory »

Details

Official Sites: Official Facebook | Official Instagram | See more »
Country: USA
Language: English | Mandarin | Cantonese | Hokkien | Tamil | French
Release Date: 15 August 2018 (USA) See more »
Also Known As: Crazy Rich Asians See more »
Filing Locations: Langkawi, Malaysia See more »

Box Office

Budget: $30,000,000 (estimated)
Opening Weekend USA: $76,510,140, 19 August 2018, Wide Release
Gross USA: $152,233,071, 19 September 2018
Cumulative Worldwide Gross: $187,451,904, 17 September 2018
See more on IMDBPro »

Company Credits

Production Co: Color Force,SK Global,Warner Bros. See more »
Show more on IMDBPro »

Technical Specs

Runtime: 120 min
Sound Mix: Dolby Digital | DTS (DTS:X) | SDDS | Sonos-ODP
Studio Time: 45 minutes

Brainstorm until each of you have 10 specific needs you’d be interested in pursuing.

Must mostly use IMDB data to provide additional value to what IMDB already does. (IMDB already looks up actors and movies.)

What data does IMDB have?

What specific needs do people have around movies?
• Think about different specific groups of people, not just yourself.

Think about the whole experience around movies, not just watching them.
• Finding a movies near you, trying to remember a movie line,

Today by 9pm - post 10 specific user need ideas on Piazza
Next’s weeks goal

• Make your own application using IMDB data and CRUD operations.
  • Define your own high level goal: a specific user need to solve
    • Define 7-10 low level goals you will need to achieve this.
  • Functionality is the most important thing.

• Must use Flask, SQLite. You don’t have to use SQLAlchemy
• We have not talked about users yet or session yet. Stay tuned!
We use iteration to minimize risk by **prototyping** the riskiest elements first.

**Idea:**
Help NBA fans recognize the strategy during games

**Paper!**  **Google it!**  **User test!**
Next’s weeks goal

- Make your own application using IMDB data and CRUD operations.
  - Define your own high level goal: a specific user need to solve
    - Define 7-10 low level goals you will need to achieve this.
  - Functionality is the most important thing.

- Create at least 1 paper prototype – test it on at least 1 person.
  - Report one thing you learned from it
- Graphic design should be GOOD ENOUGH to not repulse people in user tests.
- Test it on at least 1 person
  - Report on thing you learned

- Must use Flask, SQLite. You don’t have to use SQLAlchemy
- We have not talked about users yet or sessions yet. Stay tuned!