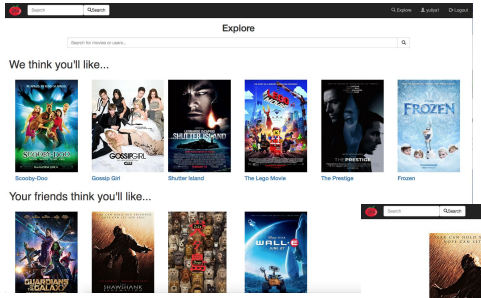
Spoiled Tomatillos

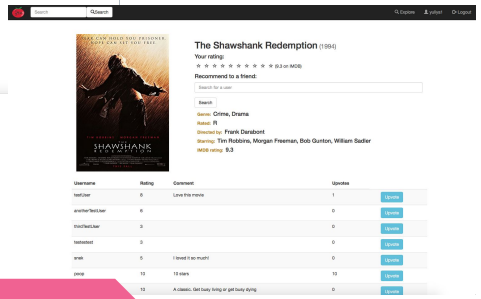
Team JANKY
Nicole Pristin, Justin Vincelette, Alice Nin,
Kathleen Newcomer, Yuliya Smilyanski
CS4500 Spring 2018

System Functionality

What did we accomplish?

- Features completed
  - create a user + log-in
  - rate & review movies
  - followers & friend-to-friend recommendations
  - system recommendations
- Achievements
  - end user use cases covered: 13/18
  - admin user use cases covered: 0/4
  - marketing user use cases covered: 0/2
- Total use case coverage: 13/24





end users were priority #1

Client Usefulness

- Client requested a social recommendation system for movies
  - Most of the requests were for end users
- We've created much of the functionality for end users
- Client can focus on creating admin + operations functionality that works best for them

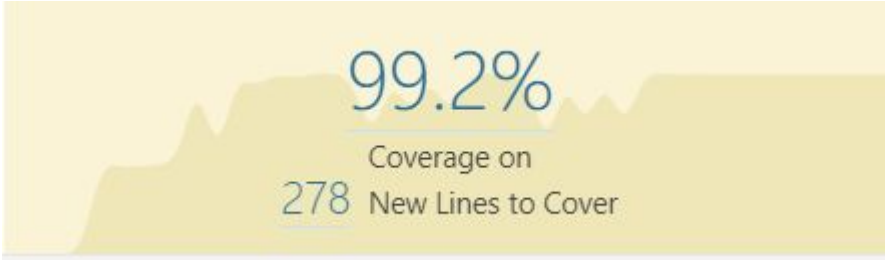
The basic idea of the system: ...

1. Users sign up for an account, and have at their disposal an IMDB-like system – that is to say, they can search for movies and learn about their plot, cast, awards, availability at local theaters, etc.
2. Users rate these movies, which automatically forms rating-based playlists and contributes to future recommendations.
3. Users can identify friends in the system – the more they and their friends use the system, the better their movie recommendations are!
4. Once a user has friends, they can then "prod" each other with movies to see (either together, or just a friendly pointer) – *shhh, we actually use these for further data collection!*.
5. Finally, the system produces recommendations via three algorithms: average critic rating (pulled from an external source), average rating on the site, and via User-User Collaborative Filtering.

Job Quality

Job Quality Metrics

- Test Coverage
  - 100% Line Coverage
  - 96.4% Condition Coverage
- Improvement over time
  - After initial testing push, consistently added more tests for new code, as well as older code that hadn't yet been tested
  - As team members became more comfortable with the stack, we were able to follow conventions more closely to produce cleaner code



Process & Teamwork

Project Management

- Kept project structured and organized by:
  - Jira - backlog, managing and assigning tasks
  - GitHub - to keep revisions straight and allow seamless collaboration
  - Jenkins - continuous integration, automating builds and running tests
- Agile team process:
  - Frequent scrum standup meetings on Facebook Messenger
  - Two week sprints
- Jenkins + Jira upkeep assigned to specific team members
  - Important to specialize

Working in a Team

- Team member roles specialized

- Shortcomings:
  - Learning curve (front-end development)
    - Javascript and React
  - Fulfilling sprint expectations
  - Communication
    - Slack-ups/ stand-ups

Technology Transfer

System Status

- Basic functionality for end users is implemented
- Code is clean
  - Good naming practices
  - Clear design
- Full stack is available on Github
- System is ready to be run by the client

Next Steps & Recommendations

- Admin user functionality
  - Enable site administration without directly using the database
  - Verify critic accounts
- Marketing user functionality
  - Bring in ad revenue
- Feed for users you follow
  - Allow users to interact more
  - Attract new users
- Increased security

Use case 17: Remove/suspend a user account
Use case 18: Edit a movie's information
Use case 19: Verify a critic
Use case 20: Removal of user from group

Use case 21: Toggle data analytics dashboard
Use case 22: Toggle between different demo user accounts