Movie Hub Final Report

Travis Bergmann
Xuanyu Li
Ranran He
Zewei Chen
Qiye Chen

04/17/2018

Overview of The Problem

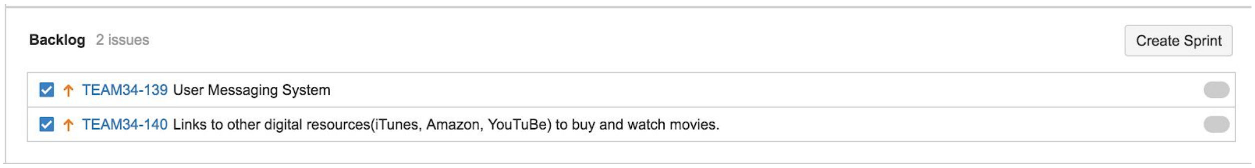
Our software group was tasked with building the first prototype release of a unique web-based movie application for the company Rotten Tomatillos. The project was meant to bridge the gap between movie database websites such as IMDb and social media sites such as Facebook. The app needed to have an intelligent design with modern features such as a recommendation system that learns what kind of movies users like based on their activity on the website.

There were several specific requirements given at the beginning of the project that needed to be included in our release. The first requirement was that users can have accounts in the system that can store profile and movie information and that can be securely logged into and out of. The system also had to have access to a searchable database of movies that contained large amounts of information about these movies, similar to IMDb. Users should be able to rate and review movies, and the ratings they give should be used by the system to give useful movie recommendations to those users. Users should be able to connect with other users and see what movies those users enjoy and have reviewed. Furthermore, they should be able to directly recommend movies to other users so that people can easily share recommendations with their friends. The main crutch of the project was that the system would intelligently recommend movies to users both by making a personalized movie list for every user and also by recommending movies by similarity to whatever movie a user is currently viewing.

There was also discussion of a few “nice to have” but non-critical features which could be added if there was enough time. For example, the system could include links to movie tickets for those movies that are still in theatres, as well as links to digital sources where users can watch the movies instantly. There was also a stretch goal of integrating other, non-movie media into the database.

Overview of The Result

In terms of overviewing the result, and based on the sprint feedback from TA as well as our own discussion, our group project was finished pretty nicely. We believe our daily slack meetings and weekly scrum meetings were essential for us keep up with the schedule. In our scrum meetings, we were able to set up fair amount of workload for each person every single week, and meanwhile checking in all the team members' progress, and help each other out if any one of us got stuck at some problem. We used Jira tickets and backlogs to keep track of our overall progress. We used vertical development techniques to ensure in each sprint review, we have all the front-end, back-end, and database setup. In terms of our completion status. We achieved pretty much all the use cases we designed at the very beginning.



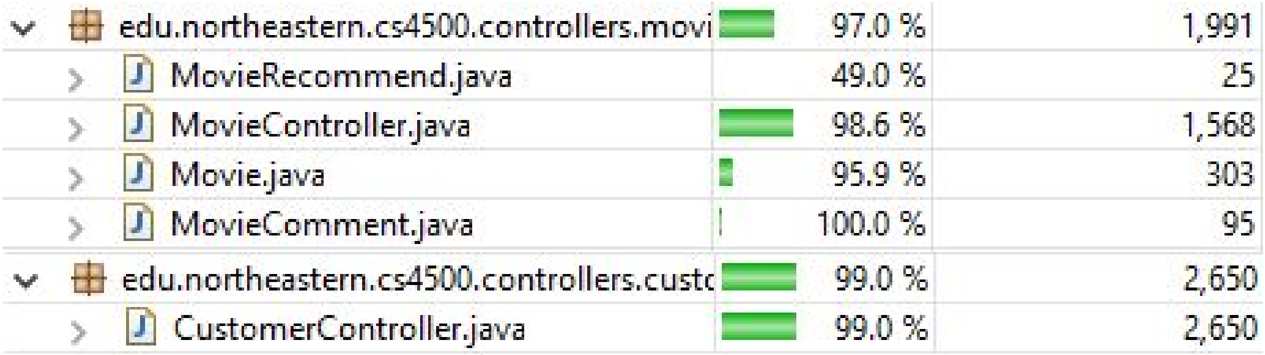
Backlog 2 issues

☑ ↑ TEAM34-139 User Messaging System

☑ ↑ TEAM34-140 Links to other digital resources(iTunes, Amazon, YouTuBe) to buy and watch movies.

Only two of them are still in our backlogs, the first one is including links to other digital movie services, such as redirecting our customers to iTunes, Amazon, and etc. The second is implementing a messaging system for our customers to interact with each other. However, the incompletion reason was mostly due to the time constraint we had, both of them are actually technically achievable if we could have another week so. For testing part, we had 99% - 100% statement coverage and over 85% branch coverage on our backend controllers and some of the

Java classes. Whereas there are a few Java classes are having low coverage which was caused by the defect of our development strategy, that we put too much efforts on creating new features leads to our tests were hard to keep up. It’s definitely a lesson that we all learned from this software development project.



edu.northeastern.cs4500.controllers.movi	97.0 %	1,991
MovieRecommend.java	49.0 %	25
MovieController.java	98.6 %	1,568
Movie.java	95.9 %	303
MovieComment.java	100.0 %	95
edu.northeastern.cs4500.controllers.custc	99.0 %	2,650
CustomerController.java	99.0 %	2,650

Overview of the Development Process

In the brainstorming stage of this project, we created a lot of prototypes, tested with real potential customers, and refined them based on all received feedback. It turned out to be a valuable input to the development of this project, as we had not only collected different perspectives that let us be clearer what we should do later on, but also it had greatly reduced the cost of presenting raw ideas that may or maynot work in the future. After that, we formulated what we wanted to do as tasks in a backlog, where some of them became active sprints in every two weeks. The sprints turned out to be a great task management technique, as everyone knew what the priorities were in a given time, so each of us could stay his/her work progress on track. While we were collaborating with each other, we divide the work of the project into different parts, where no two of them overlap and each of us took one of them as his/her responsibility. It again turned out to be a great task management technique, as it greatly reduced possible conflicts that teammates might have in the development process. One more process that worked well was the utilization of Git. Previously almost all of us wrote source code in such a manner that when we wanted to change something, we simply replaced things we didn’t want by things we did want, leaving the previous versions of the source code lost; however, new changes were not always better than old versions. With Git, we were able to keep track of everyone’s source code written in any given time, compare different versions of source code, and even revert the source code to a previous version if the new versions was found to be not working.

Since we were still inexperienced developers, we faced some difficulties with some of the development process that we chose. At the very beginning of the task management, we didn’t partition the work of the project well enough that some of the work had overlap, but each of us still took responsibility on his/her allocated assignment. Because of that, there were few times that our work had conflicts when we tried to merge them as a whole. One of the examples was the joint work by Ranran and Travis, for developing the web pages that render movie details, in earlier sprints. When conflicts happened, the team could only accept either one of their work, which might leave compatibility issues in the future. Realizing that even for the same thing, each of us could have different styles against it, we learnt that the best approach for collaboration is to partition the whole work into disjoint sets of work, provided that everyone can still help with each other whenever there are difficulties.

A Brief Retrospective of the Project

We enjoyed the developments on each stages, although it was often annoying when bugs occured. Everytime we added new functionality, we were proud and excited at our project's growth and improvement again. We also love challenging ourselves and this project presented us with plenty of stimulating challenges. We set a high standard for ourselves, even though we are not familiar with some of the technology and work that needed to be done. We kept refining every aspect of the project including functionalities and UI designs especially when it came to security and user friendliness. During the whole semester, our back end members continuously analyzed and came up with more efficient ways to get work done, and our front end members kept adjusting UI designs to make our website more beautiful and easier to read. There wasn't much to dislike from the project, all the work seemed reasonable and useful.

We have contributed a lot to the project and of course also learned a lot. We have a clearer idea of how to make website security, how to provide great user experience, how to conduct a UI/UX interview and test, how to improve designs based off feedback from the interview, how to connect front end and back end efficiently, how and when to write effective test suites, how to manage our time, how to divide work properly, ect. The project gave us exposure to a complete process of how to do a team project. The sprint system is a helpful way to manage the process of a project and the team meeting of each sprint did a great job providing us with feedback and helped us decide what to do next. The experience learned in this project will help us a lot in the future studying or working.

During the project, although we didn’t run into too many painful difficulties, sometimes it would be very difficult if a team member was not familiar with html, web development structures, or any of the other project-specific technologies. It would be helpful if more materials related to this were readily available so that people who don’t have web development experience could get used to the project more quickly.