

March 14, 2018

**From:** Team-34

**To:** Rotten Tomatillos Client

**Re:** Project needs and team experience

**Xuanyu Li**

Phone and Email:

During the development part, we implement the phone message function and send email function. We are able to send the email and message to customer when they create account or try to find the password. This part does not stop us a lot, because I did those things which were similar with my 2nd co-op. I finish those two functions very fast. We still need to create more functions by those two methods. For example:

- Send the customer recommendation movies,
- Customer are able to unsubscribe those notifications,
- Finish the phone verification.

The basic function for sending message for the phone and email are done. I still need more time to implement the methods for receiving message and analyze the result from the customer.

Test:

I did not test any code with the spring boot and the tomcat. I did not know how to use the Mockito by Spring boot, therefore, the most part of my code is untested. I discussed with Qiye Chen last Tuesday to talk how to use the Mockito test and JUnit test, because he had some experience about the Mockito test. I will test those code in a few days.

## **Ranran He**

### User Profile:

User Profile now can recognize which customer is and show his information. We now have already had follow and unfollow functions. Other parts are basically done in front end and waiting for back end, for example: create/delete/modify Movie playlist, change password, and change basic information. We have not started “send messages” part yet. This will also be one part that we need to work on.

### Search:

Search part has been done, we now can give the results which can be divided to “user” and “movie” categories, user can check only user or movie results by clicking according tabs. In the future, we might add some more categories or functions to improve user experience.

## **Qiye Chen**

### Back End Design:

Currently the back end consists of just one layer, where the RESTful APIs and database CRUD operations reside within the same layer, and one RESTful API just corresponds to one database CRUD operation. In earlier times of this project, there seemed to have no issue since one simple database CRUD operation typically was just what a RESTful API needed to do at all, such as get user by ID. However, in this sprint when we were developing more APIs, such as get all users followed by a particular user, we found that we also needed to utilize the same database CRUD operation used in the get user by ID API. In short, we realized that one database CRUD operation can be utilized by two or more APIs. If we continue keeping them in the same layer, then things can get complicated in the future. Therefore, in the next sprint we are considering separating the APIs and database access into two different layers, trying to make things organized and efficient.

## **Zewei Chen**

For this project, I am mainly in charge of the front-end UI design and controllers. Meanwhile, adding needed frameworks, maintain AWS (Jenkins, docker and other infrastructures), and troubleshooting for the team.

- User Login / Registration (Front-End): I used my knowledge from web development course, implemented the UI and controllers for this feature. The reason I am working with front-end, is simply because it's my weakness, and I am regarding this project as a learning opportunity.
- User Follow / Unfollow Functions (Front-End): Again, since I developed a similar social platform project for my web development course, so I got experience of implement this feature. By using angular and session.
- Integrating Angular: Our team chose to use Angular as our front-end framework. Since I took the web development course a year ago, so I worked with my teammates to add Angular to our project, then later I was able to create and setup file structures for the team myself.
- Integrating Session: We used session to store user-login information in the front-end. Since the STS project has different structure from the one I did for web development course, so I did some research, and found sessionStorage to resolve this issue.
- Maintain Jenkins and Docker: I was using Docker in my first CO-OP, Jenkins and Pipelines for my second CO-OP. The free-tier EC2 brings up issues so often, but I was able to use my CO-OP experience to resolve them.
- Design and Debug: I am a fifth year CS student, I have taken OOD, Web Development, Database Design, and other courses. I consider myself an experienced student. So occasionally, I helped teammates to catch and troubleshoot bugs in the code. Also, provides ideas of implementing data and model structures.

## **Travis Bergmann**

I have a lot of experience with front-end scripting using AngularJS as well as with making html frameworks. I frameworked and did the initial scripting for the search page front-end using the knowledge and experience with AngularJS that I gained during my first co-op. Furthermore, I have created a rating and review front-end system by scripting html that had been created to allow users to submit and display reviews. Again I pulled from my knowledge gained during co-op to complete this. I do not have much

experience with CSS styling, but this gap seems to be filled by Ranran who has been styling most of our html. The front end scripting has also been done by Scott so it is fully covered. I also have some experience with backend API coding which has helped me understand how to get my front-end functions to communicate with the back-end, but this has also been covered by Xuanyu and Qiye who have written the actual back-end code. I have a lot of experience with database design, and so do Xuanyu and Qiye so that has also been covered. We all have experience writing tests so we have sufficient expertise to create test suites. I have also been in charge of several code reviews and completed many pull requests, especially those related in some way to the front-end of the project.

There are only a couple potential gaps I can see in the project. First is that none of us, as far as I know, have extensive knowledge of web security so we may have to study how to prevent XSS attacks and SQL injections and then implement that code. Also, none of us have much machine learning knowledge, so we may have to research how to make an effective engine for movie suggestions.