

Issue Tracker Express

SPRINT RETROSPECTIVE 2

CPSC 3720

UNIVERSITY OF LETHBRIDGE

Team Aegir

Lorenzo Conrad

Mathew Richards

Steven Deutekom

November 16, 2019

1 Introduction

We have completed our second sprint for Issue Tracker Express. Below we reflect on the experience in order to improve our process for the next sprint. We will give some details of our initial plan for the second sprint. Then we will show what we actually got done. This will be followed with some description of a few of the problems we encountered during the sprint. Then we discuss a plan for the third sprint and describe what we will try to do differently the next time around.

2 Sprint Plan

The intention for our second sprint was to make up for missing features from sprint 1 and still get our original sprint 2 goals finished. This meant making it possible to add and view custom issues, and users. Previously we were only able to view all issues and users, but they were predefined. We also wanted to make it possible to fully update and delete an issue or user. We hoped to get all of these features working both on the server and on the web client. This would give a potential customer a good idea how the system would function with issues and users so they could test it out.

For the web page we first wanted to get the forms for adding issues and users to accept custom input. Once this was working we intended to add pages for viewing a single issue or user. This meant adding links to individual issues and users on their respective view all pages. When clicked these links would open a page to view all details of issue or user. While on this page it would be possible to delete the object or to edit its

fields to update it. We expected to open a form similar to the add issue or user forms that had the current values of the object in it. These could be updated to reflect the new desired values. After this viewing the single object would reflect the changes.

On the server side of things getting these things to work meant adding a new endpoint for DELETE to issue and user endpoints. It also meant adding some functionality to the handlers to deal with different parameters that would tell the server whether to return a single object or a list of objects for issues and users. We did not intend that there should be too many changes to our model classes to accomplish this. Most of the updates would be in the handlers that our server ran for different endpoints.

We planned to continue to have weekly meetings to discuss our progress. At these meetings we could each explain how we were doing with our individual tasks. We also would talk about the next steps we could take and what our schedules looked like. At these meetings we could take notes on our potential designs and keep ourselves on track. We also gave ourselves the opportunity to help each other solve any problems that may have arisen.

3 Sprint Results

At the end of our second sprint we completed our goals in a slightly different way than we expected. The first thing that we ended up doing was refactoring our server and handlers a bit. This was largely due to the issue testing with g++9.1 not working properly. Because no code that has restbed included can be tested without g++9.1 we were not testing our server and handlers except with the web page. To get around this we decided to move as much of the functionality as possible into the model classes. This turned out to lead to a better design anyway.

We accomplished this by writing some extra functions for creating objects in our IssueSystem that took in JSON strings. These functions parsed the body of a POST request and used it to create the desired object. We also used this same approach for updating objects using a POST body JSON string. During the second sprint these functions will be adjusted to use the same code since they are so similar and they will also enable the loading of server state saved in JSON format. Most importantly this moved all work for parsing JSON and creating objects out of the handlers so it could be tested.

On the server side we were actually able to get all the handlers and endpoints set up to do everything that our server is supposed to do. There were a few small things that will still need to be done, but the server and its system are mostly finished. We have handlers and system code to work with comments and most of

the code to filter issues by priority and status. The refactoring we did at the beginning of the sprint made it much easier to add functionality to the server and handlers, which made server side work much easier to complete.

Unfortunately, on the web client side we were unable to get all the functionality for these feature done. We did manage to get the forms to accept custom input. We also found a better way to open the forms so that when they submit the page refreshes back to the view all page and shows the newly added objects. We also managed to get the single view pages to work. It is now possible to click on the title to an issue or user and get a page with more detail. For issues this will be where the comments for that issue are displayed once implemented. We did add a delete link to these pages, but unfortunately we could not get the delete request to function properly with the web page. So currently it does not work. Because we got most of the server side functionality done for this sprint we expect to spend most of our time working on the web client in sprint 3 to make all of the functionality available.

When it came to communication and meetings we did worse in many ways than sprint 1. We did not always hold our weekly meetings and we only got together in the lab a couple of times before reading break. Luckily the sprint was extended over reading break. We were all able to be in the lab to work out what needed to be done. We did manage to communicate to each other that our schedules were too busy and eventually find some time to meet. We see each other often and use slack to keep in touch when we do not. So there were some successes in dealing with the fact that we did not have the time we wanted to meet and work as often.

4 Problems Encountered

Our main problem this sprint was getting the JavaScript request for delete to be sent. Coupled with the inability to test handlers and server code properly this made it difficult to tell if the problem was with the JavaScript and web browser or with our server. Fortunately Steve spent the summer working with python to make request to various APIs. This enabled him to write a quick script in python that called each endpoint and ensure that they were working. This script helped us confirm that the server was working just fine for all of our endpoints. This also gives us a nice integration test that automates calling all our endpoints rather than having to open up the web client to test the server. As we build the server we can use the python script to ensure it still functions properly and testing the web client will only be required to test web client functionality.

We still had to deal with the issue with testing needing g++ and restbed needing g++9.1. However, with the redesign of our handlers and the addition of the python script we have managed to get very good coverage of our code anyway. It was not a bad thing to run into because it forced us to think about how we could automate testing as much of the code as we can. It is always good to think about what kind of design best facilitates testing, and also to know that there are many ways to automate testing if your testing framework cannot do everything you need it to.

5 Plan For Next Sprint

The next sprint is the final sprint so we will need to finish everything we have not done yet. This will mostly include finishing the web client so that it uses all our systems functionality, as well as saving system state. If we run into problems with JavaScript we may have to adjust some of our endpoints to types we know work. But with our new design this will not be hard at all. We will also need to fully test and confirm that our back end is complete. There are a few things, like when a user is removed updating all issues and comments that link to them, that need to be addressed. And once this is all finished we will need to write up our user documentation for the web client and server. Finally, we can just take any remaining time we might have and polish the client and server.

6 What To Do Differently

Just like last time the major thing we can do differently is try to be better at scheduling specific times to meet and tasks to do. Because we see each other often it is not hard to communicate, but we have not done a great job of specific planning. We know that with the semester coming to a close soon there will not be any let up in projects that need to be finished. So it will be important to set aside the time we need or we risk not being able to finish everything (there will be no reading break to extend the sprint into).

We will also need to communicate some of our ongoing technical issues with Dr. Anvik. The main one being our JavaScript issues. If we can get some experienced help we may be able to discover what the problem is with how we are making these requests, or at least learn how to better debug them. It is always good to work to solve our own problems and gain experience, but when there is less time it is important to get the help we need to deliver a quality product on time.

We have also decided to adjust our issue filtering feature. We originally thought it would be cool to filter in many different ways. However, due to time constraints and complexity we are going to limit this feature

to filter by priority, status, and a single tag. That way there is a reasonable amount of power for the user to narrow the search for issues to complete, but still a manageable amount of complexity. These features would not be too hard to expand upon after the project was delivered if the customer needed it. We imagine there would probably be some patches and feature adjustments expected after the system was shipped anyway, so we could roll the expanded filtering into those.

7 Conclusions

We started this sprint off slowly and were a bit worried that we would be leaving ourselves too much for sprint 3. However, we were able to focus our efforts onto the server side work and make really good progress there. This leaves slightly less functionality for a customer to see on this sprint, but puts us in good shape for finishing all the features in the next sprint. We also feel good that we have been able to work around some of our problems in practical and efficient ways. Different methods of testing and some design changes have helped us produce a system that is more fully tested and of higher quality. We have continued to use our Kanban board and try to engage in agile practices, but we feel that these areas could be improved. It is our first time working with these concepts though so it is probably expected. Also, when we were learning about Agile it was clearly stated that it could take a while for an organization to adopt it. As we are just starting out it is not surprising then that we have had some trouble sticking with it. Overall the experience has been positive for all of us. We look forward to the final sprint and finishing with a good product. We have learned a lot so far about our software design and the software engineering process. Plus all of the issues with dependencies not working together has provided valuable insight into the challenges we will face in the real world.