APPSO Portal Monitoring and Control Metrics from Sprint 3

Key measurables and metrics:

Efficiency: Minimal Waste; Maximum Throughput

Compliance: Degree to which a program starts **providing value** to the customers

Correctness: Degree to which a program operates according to specification

Integrity: Degree to which a program is **impervious** to outside attack

Usability: Degree to which a program is easy to use by the intended audience

Maintainability: The degree to which a program is amendable to change

Team Objectives for maintaining such metrics:

Efficiency: Velocity of 150 story points per sprint

* Why 150 story points? This is our last sprint before UAT and Turnover, so we needed to increase the number of story points completed this sprint to reflect the remaining user stories left.

Compliance: 100% of the **most valuable** user stories deployed

* This is a non-negotiable, for us, by sprint 3, we should have implemented all of the most valuable user stories that help complete the **functionality** of the website

Correctness: 0 errors **escaped** to the user

* We defined 0 errors escaped to the user as: No Errors visible in the **production** app.

Integrity: 0 security breaches

* We check for security breaches by continuously running brakeman while we are implementing user stories and during CI/CD GitHub Actions tests. We also create integration tests that test how secure the application is from malicious individuals from accessing pages they are not allowed in.

Usability: Intended user learns how to use the system in 15 minutes or less based from a set group of Acceptance Criteria that adheres to how usable the system is.

Maintainability: Mean Time to Repair (MTTR) - from assignment of ticket to deployment of fix – of 5 hours.

TEAM OBJECTIVE RESULTS (to be filled out throughout the sprint 3):

Efficiency:

On the last day of Sprint 3 (11/14), the number of story points we have completed this sprint is: **217 story points!** Our team did an amazing job of finishing up any remaining or additional user stories. We managed to produce a complete product that the customer can use (pending UAT Acceptance). The sprint burndown chart below shows the progress we made over time.



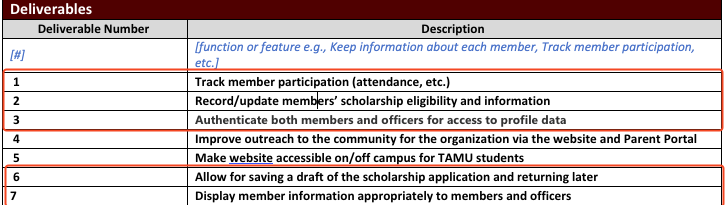
Note: we marked off several user stories at a time, especially during/after SCRUM meetings.

There were some stretch goals included in our Jira product backlog, but we deleted them because these were additional features that were included in Project **Deliverable and therefore not valuable.**

Compliance:

Most Valuable User Stories needed to complete:

From the project scope:



These are the most valuable stories we need to complete this sprint:

“As an APPSO member, I'd like to be able to see my personal participation stats tracked in an easy-to-read format so that I can know how to improve my standing.”

“As an APPSO officer, I need to be able to see specific and overall member rankings based on various stats so that I can make a data-driven decision on the winners.”

“As an APPSO officer, I need to be able to visualize members’ involvement rankings, so that I can help decide scholarship winners.”

“As an APPSO member, I would like to be able to see my own application completion percentage, so that I can budget my time for completing it.”

“As part of the APPSO Scholarship Committee, I would like be able to see members' application completion percentage, so that I can remind them to finish applying.”

“As part of the APPSO Scholarship Committee, I would like to view completed scholarship applications, so that I can start to make a decision on who is the scholarship winner.”

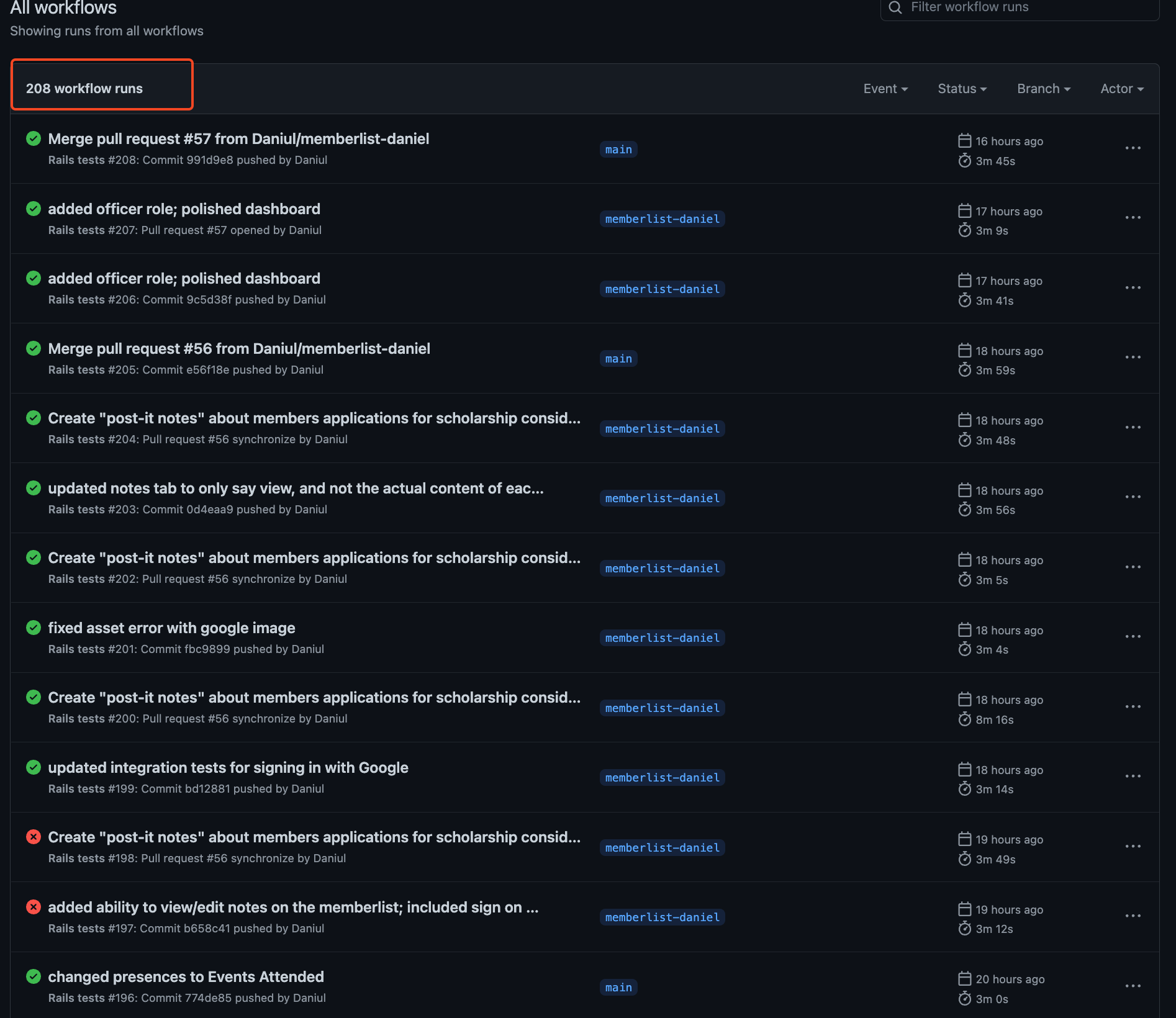
These user stories are mainly pertaining to scholarship-related functionality in terms of viewing submissions and taking note of eligibility and potential winners via a list of users.

**Green represents the most valuable stories we have completed.** We managed to complete all our most valuable user stories, and we are proud of that. **We achieved 100% deployment for our most valuable user stories.** During the UAT phase, there could be small changes the user would like to make regarding the implementation of some user stories, but as of right now, we have **fully completed** all the most valuable user stories.

Correctness:

GitHub CI/CD Tests throughout Sprint 3:

<https://github.com/Daniul/APPSO-Parent-Portal/actions>



Continuing and finishing our progress front Sprint 3, we still managed to have **ZERO** errors that escaped to the users, thanks to the code reviews we conduct during open pull requests, and CI/CD integrations tests shown above that are always conducting during updates to branches and open pull requests. We leverage these tools in order to catch any errors before they reach **production**.

We did have sometimes during **development** that errors would pop up to the user, but we were very prompt in fixing such errors, which mostly entailed creating/updating tests to reflect changes to the website.

https://github.com/Daniul/APPSO-Parent-Portal/pull/49

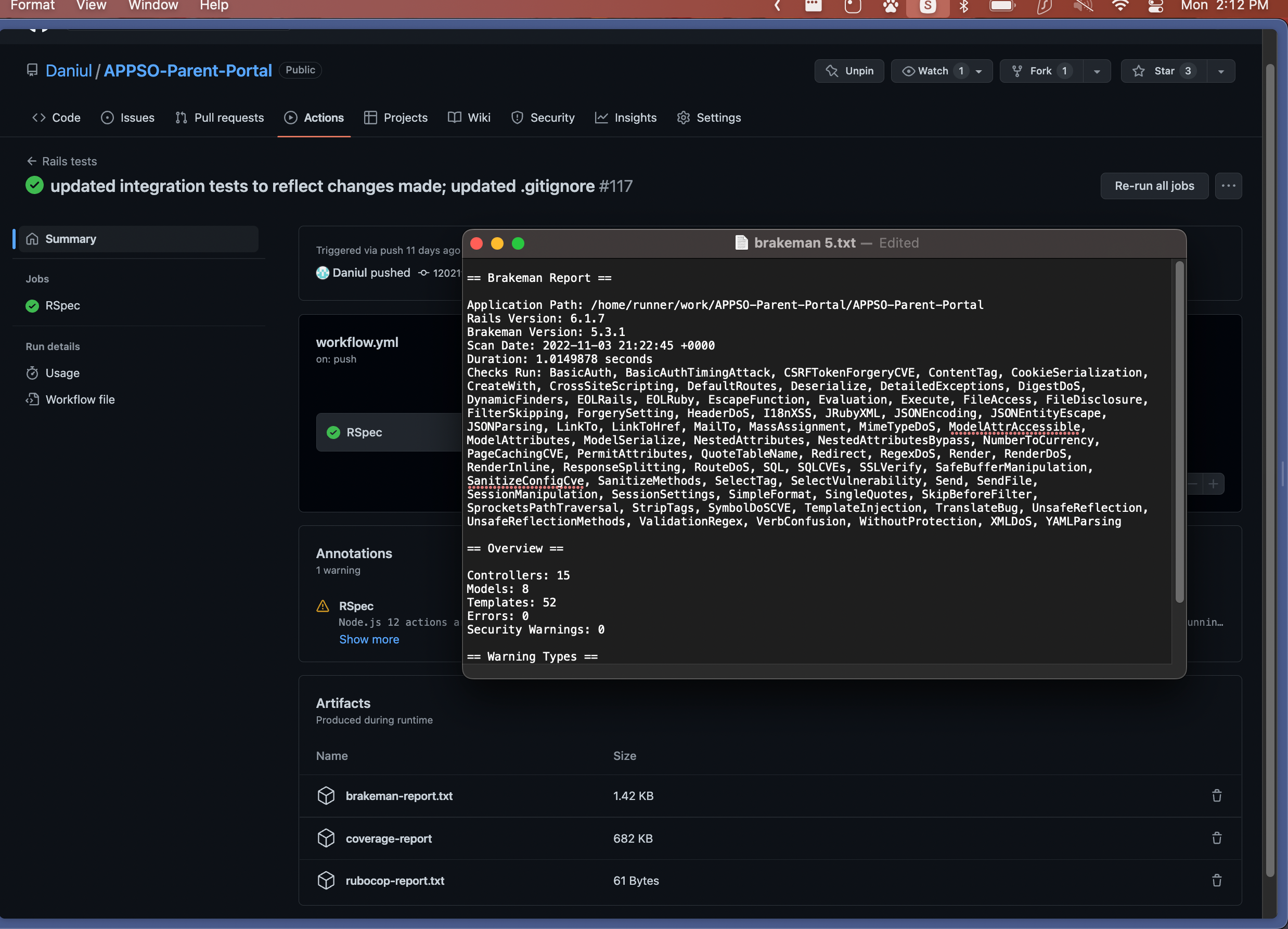
This pull request is a **great example** of how a user story/feature was tested against the acceptance criteria, and the Definition of Done. Thierry was updating and polishing the scholarship user stories and updated the question model. The unit tests made for the questions table were now out of date, and now reported an error. This was **caught in the GitHub Actions CI/CD** test, and Thierry was able to update the unit test and get the CI/CD checks to pass. Utilizing **the Acceptance Criteria and the Definition of Done allowed us to ensure that the program operates according to specification**.

**In conclusion, as of the last day of the sprint, we do have zero errors (seen so far) escaped to the user**.

Integrity:

Compared to Sprint 2, we did not have many serious security vulnerabilities that were produced by Brakeman or missed by normal acceptance criteria and integration testing. Brakeman continued to show no potential vulnerabilities throughout Sprint 3. Below are a few examples:

Exhibit A:



Updating Integration Tests to make sure its still secure:

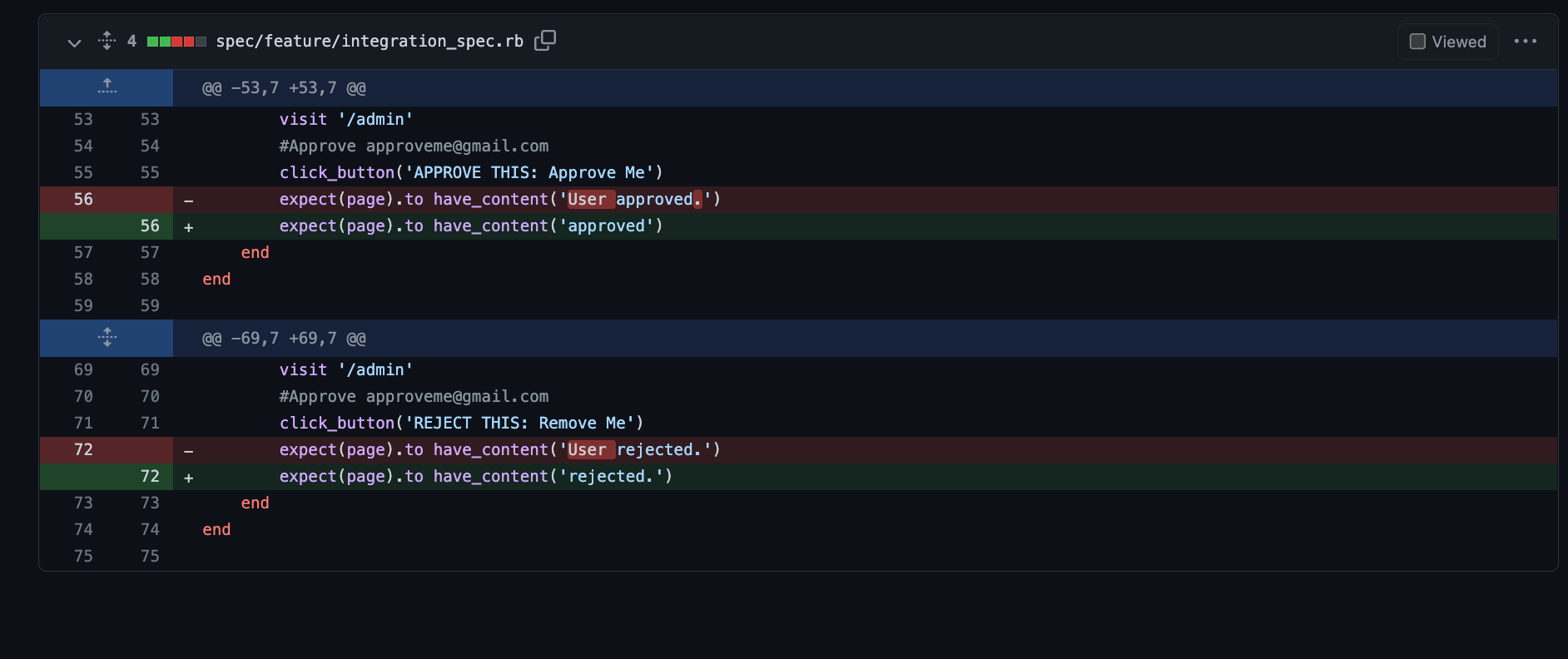


Exhibit B:

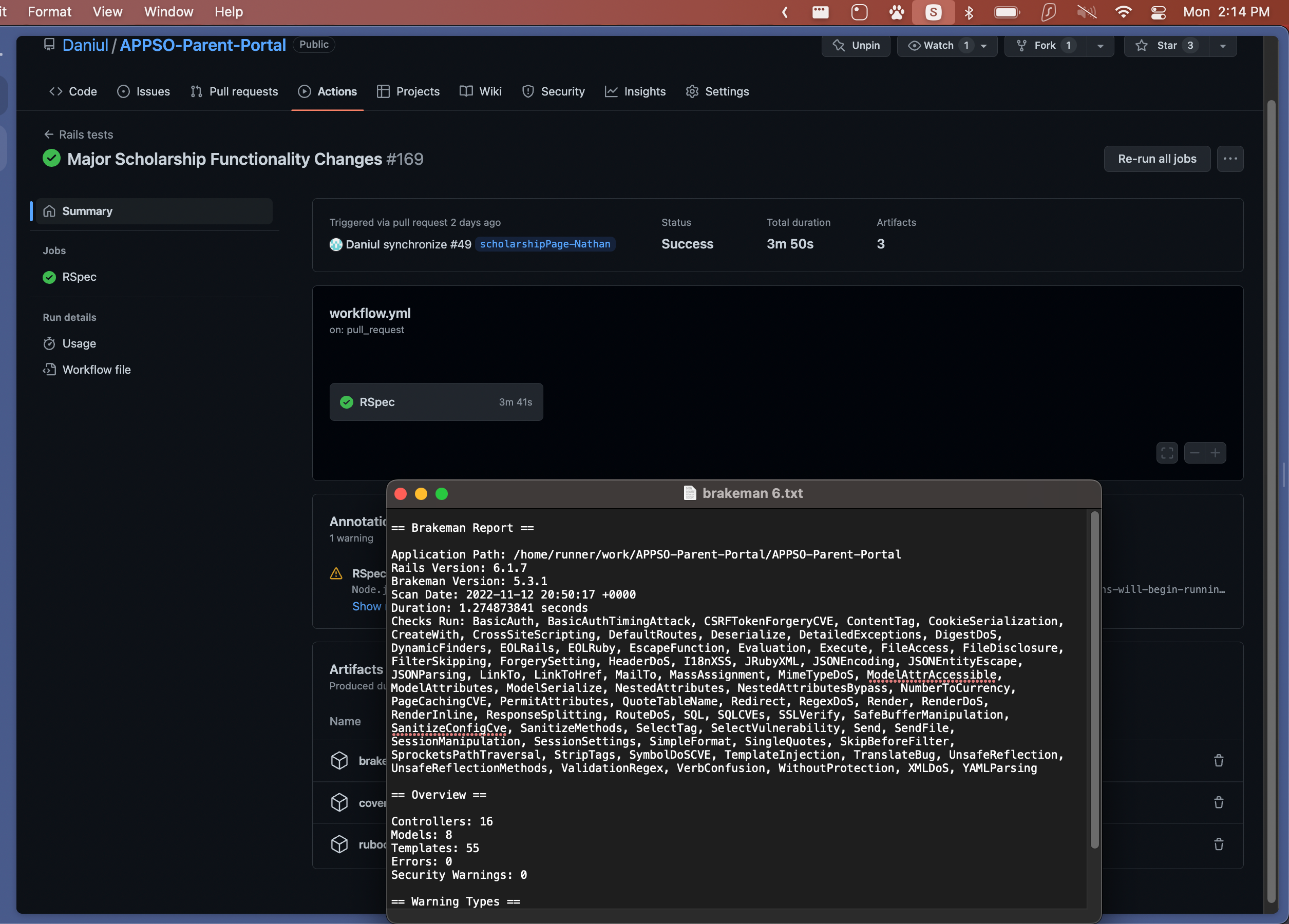
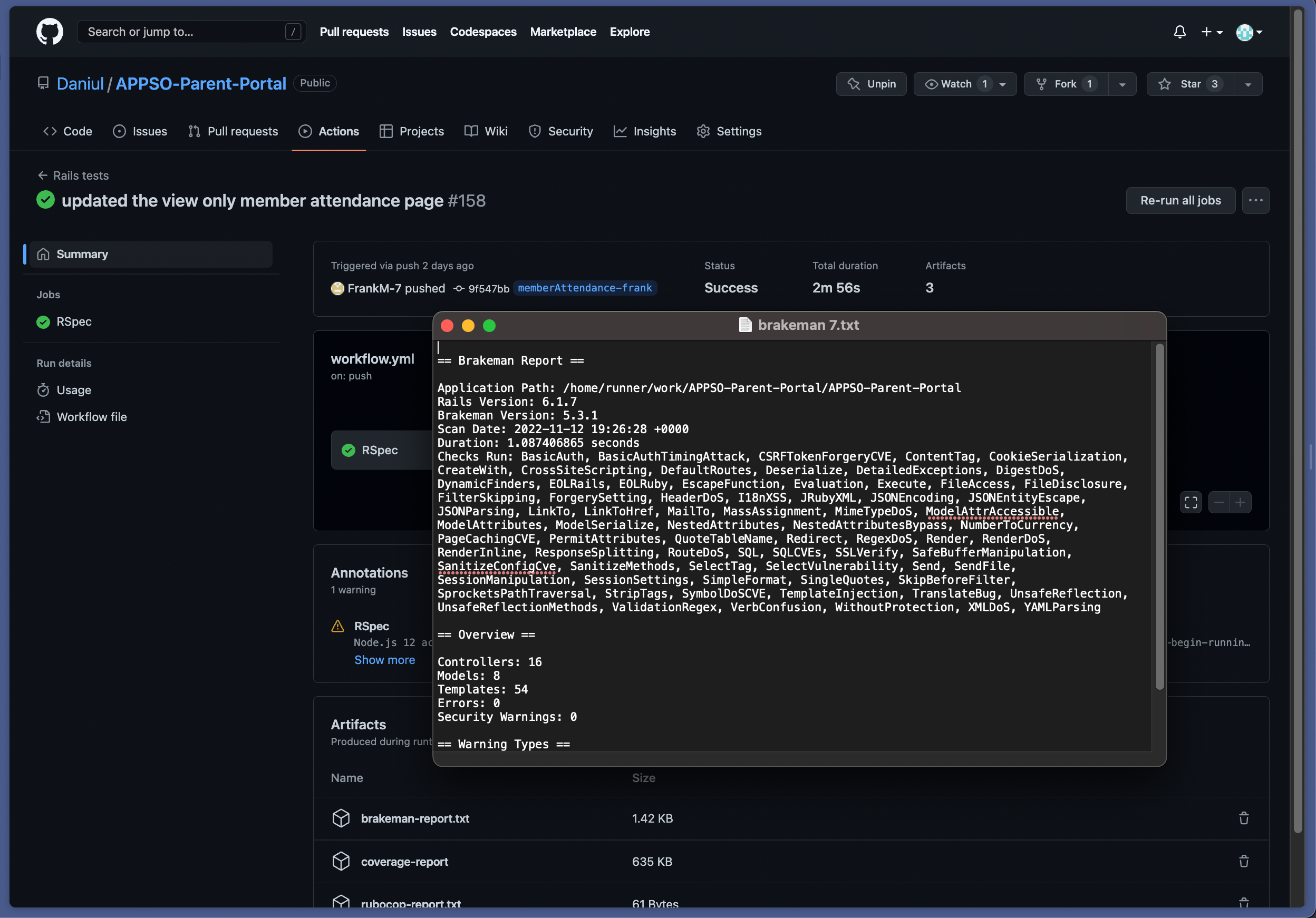


Exhibit C:



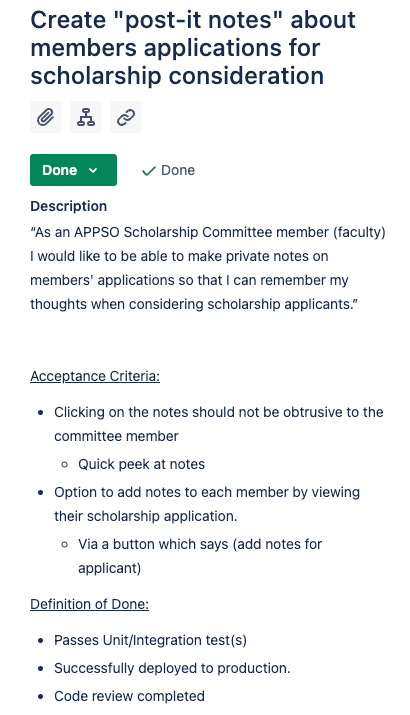
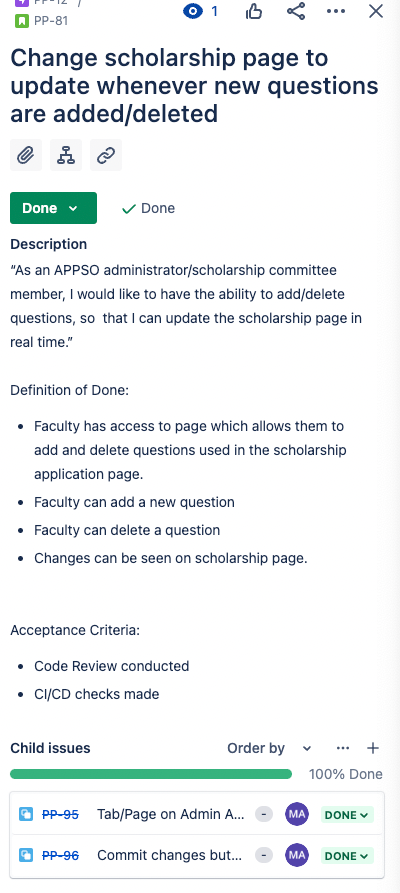
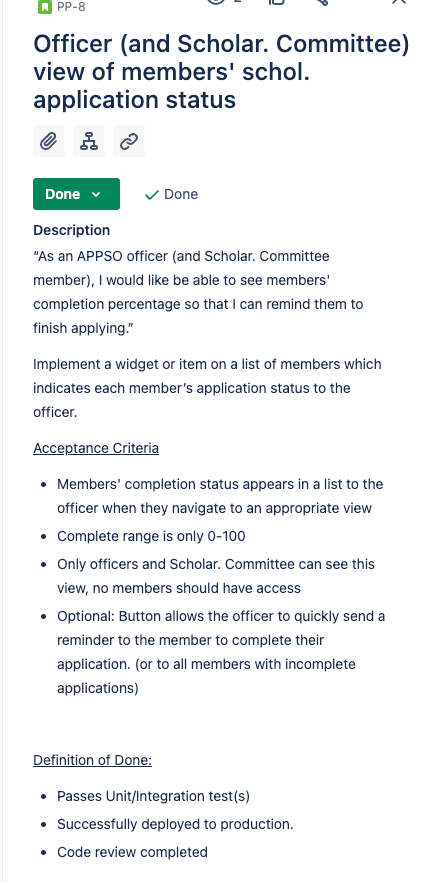
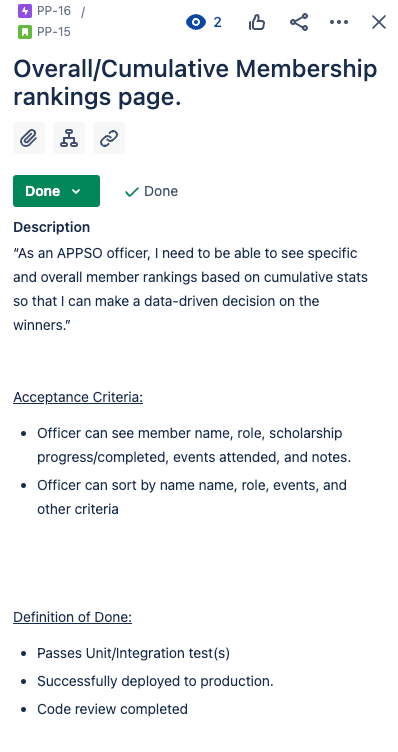
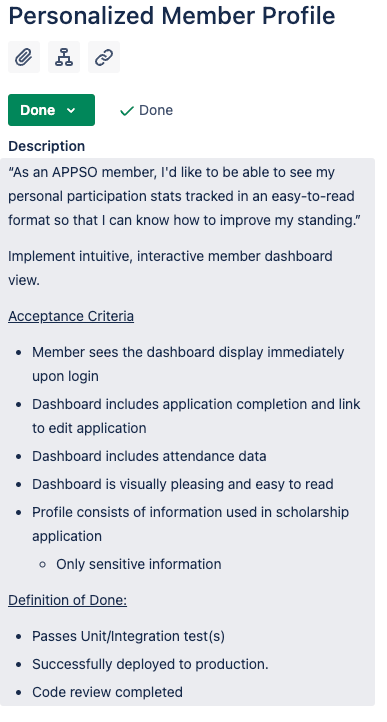
So as of the end of this sprint, we have **ZERO** security vulnerabilities

It’s great that we managed to have zero vulnerabilities at the end of this sprint. We will be looking and monitoring brakeman warnings as they come.

Usability: Intended user learns how to use the system in 15 minutes or less based from a set group of Acceptance Criteria that adheres to how usable the system is.

11/14: Thierry asked roommate to interact with the website once again.

Thierry: I once again asked my roommate that I tested on last sprint, gave him the following list of user stories and acceptance criteria:



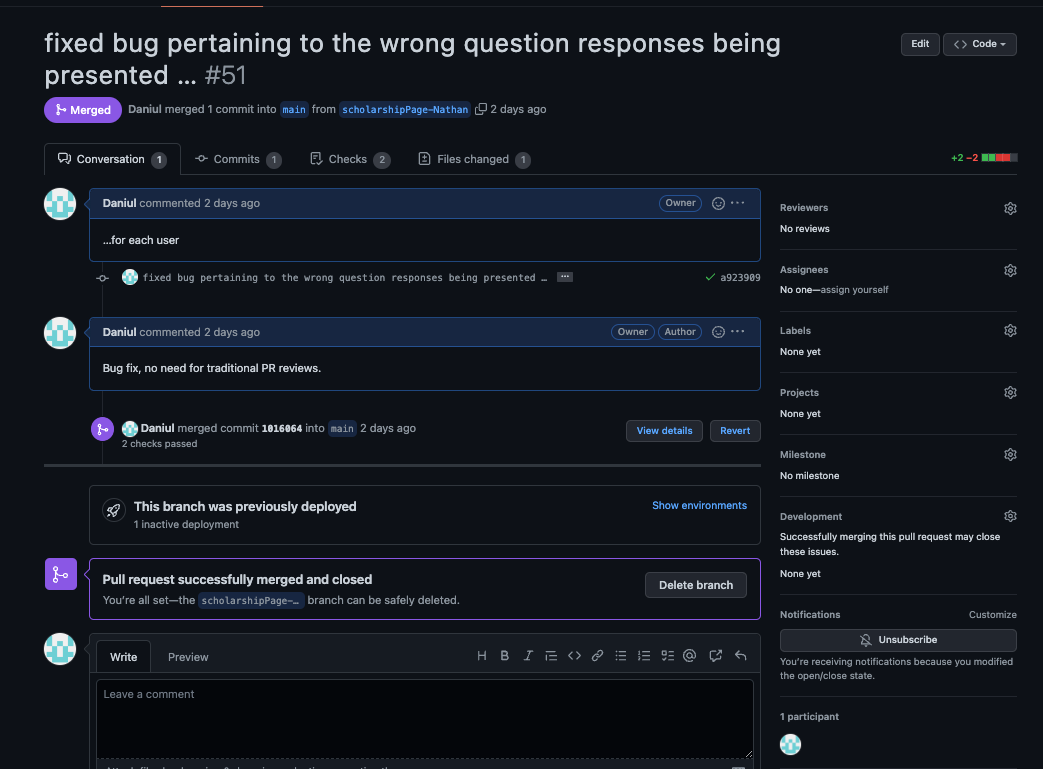
These were core user stories that reflect on how usable our application is. We timed his time of acclimation to 15 minutes or less. After distributing the user stories and acceptance criteria to him, my roommate was able to complete all the acceptance criteria at just about 8 minutes.. Keep in mind, my roommate is a Computer Engineering Major who does have experience with creating web apps, so he was a bit more acclimated to the system than a normal student.

This time metric of 15 minutes or less to get used to the system **will vary, especially with real parents and members of the organization**. Some users may not be as tech savvy as others, variety matters.

**But our test run with Thierry’s roommate confirmed to low time to acclimation.**

Maintainability: Mean Time to Repair (MTTR) - from assignment of ticket to deployment of fix – of 5 hours.

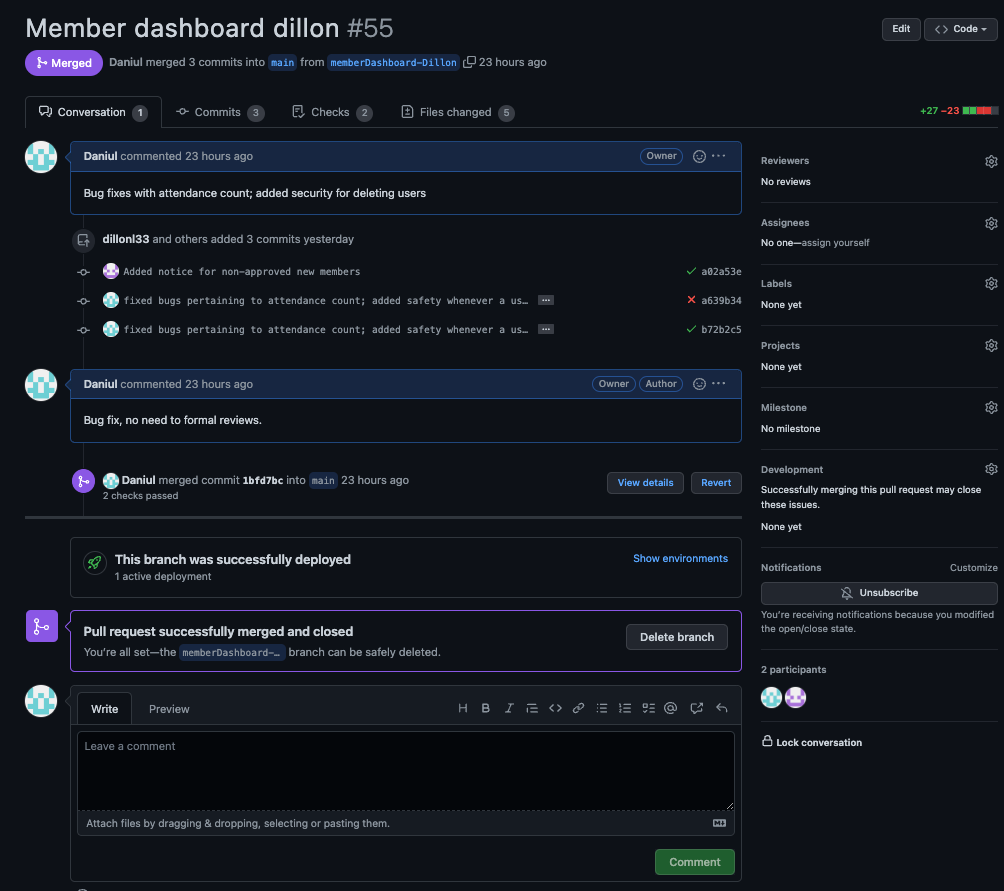
**BUG FIX EVENT #1: Google sign-in button broke website**



From Assignment of ticket to deployment of fix: ~30 minutes.

Thierry: As mentioned earlier, I was the one who helped fix this error, it was a quick fix after examining what was in the src attribute (filename did not include file extension, which was required for heroku).

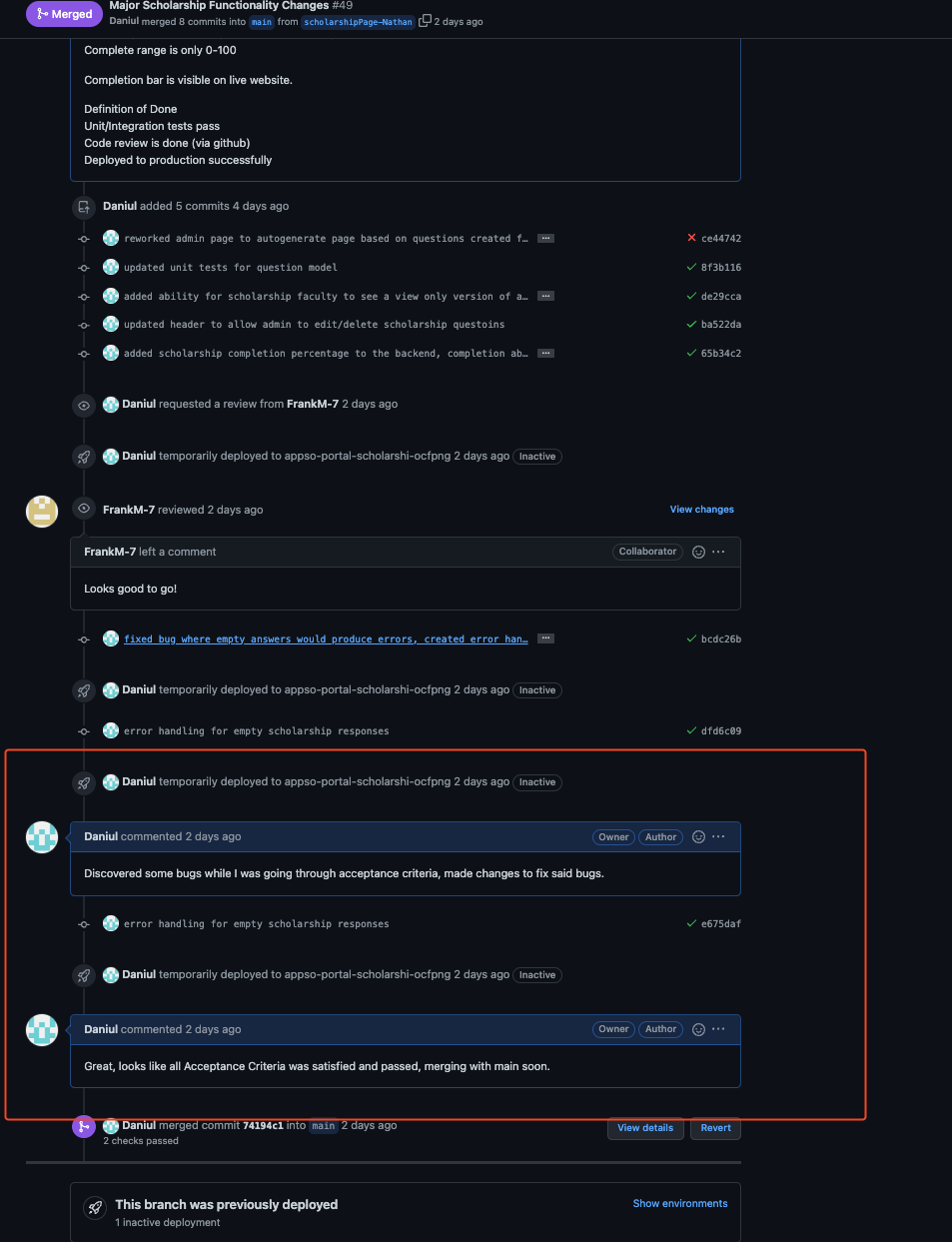
**EVENT #2: Attendance Count was not correct for some users (Was counting deleted events);User cancelling their accounts did not delete their event and scholarship data)**



From Assignment of ticket to deployment of fix: ~1-2 hours.

Thierry: A bit lengthy bug fix, I noticed one of the implementations for the member dashboard, the attendance count, would report the wrong figure and create inconsistency, but while fixing the bug, I forgot that we didn’t account for deleting a user’s event and scholarship data when they requested to cancel their account, so I also included that. These can be two separate bug fixes, but I fixed them in one pull request.

EVENT #3: Fixed errors with empty scholarship responses



From Assignment of ticket to deployment of fix: 15 minutes

Thierry: This was discovered while Frank and I were code reviewing my changes, and testing it via the review app. I noticed that the scholarship responses for users where it was empty would create an error. I quickly fixed it by creating a default value of “None”, the actual text, for users where responses were empty/had only whitespace.

The above are 3 events of MTTR. We had other errors /scenarios where we had a ticket to fix them, but they are denoted in our commits.

**SO MTTR was around 1-2 hours, I would say 1 hour usually since we were always on top of things when it comes to detecting unexpected errors.**