Testing report

Acme AirNav Solutions, Inc

Group Number: C1.066
Repository: `https://github.com/mquirosq/Acme-ANS-D04`

Member:
Daniel Herrera Urbano, danherurb@alum.us.es

May 26, 2025

Contents

Executive Summary

This document contains a test report on two mandatory requirements for Student 3:

- 8. Operations of Flight Crew Members on Flight Assignments.
- 9. Operations of Flight Crew Members on Activity Log Records.

The tests performed followed the methodology suggested in the theory lectures, as well as the procedures shown in the review sessions.

Revision History

Revision	Date	Description
1.0	2025-05-26	Initial draft
1.1	2025-05-26	Changes in performance section

1. Introduction

To verify that the features implemented in the third delivery were correct, testing must be performed. This is useful not only for us developers, but also for our customer. In the end, it is another way to prove the reliability and robustness of our system.

The following document is divided into two main sections, each one regarding one of the mandatory requirements proposed for testing. Furthermore, each requirement will be divided into subsections regarding the available features they provide for an entity, which are the listing, showing, creating, editing, deletion, and publishing of the former.

2. Testing on flight assignments

Flight assignments had seven features: list completed, list planned, show, create, update, delete and publish.

Listing (list completed/list planned)

Because the two listings are really similar, they were tested at the same time.

First, tests performing legal requests were performed. For that, I signed in as every flight crew member available and accessed the two listings from the menu.

Secondly, tests performing illegal requests were performed. For it, I tried accessing to the both listings from an anonymous user and a customer user, both giving authorization errors. There was no way to hack the URL in order to obtain another's member listing, so this type of hacking was not contemplated for the testing.

No issues or bugs were encountered during the recording of these testings.

Show

First, tests performing legal requests were performed. For that, I entered into every flight assignment in both listings for each flight crew member.

Secondly, tests performing illegal requests were performed. For it, I tried accessing a flight assignment of member1 from another flight crew member, another realm and an anonymous user, which resulted in an authorization error. I also modified the URL in order to check the system response to a show request with no ID parameter, to ID=-1, to ID=, and to ID=XXX, for which I received authorization errors.

At first, a request with ID=-1 was authorized but resulted in an error coming from a flight assignment which could not be found. Because of this, the auth() method in the show, create, update, publish and delete services had to be changed. After that, no issues or bugs were encountered during the recording of these testings.

Create

First, tests performing legal requests were performed. For that, I entered as member1 and created a new flight assignment from the ”Create” button in the listing. I first sent the form empty, and then I tried every possibility for each field with the facilitated sample data obtained from the ”Scrapbook/Sample-Data.xlsx” file. Finally, I sent the form with valid data and successfully created the flight assignment.

Secondly, tests performing illegal requests were performed. For it, I tried creating a flight assignment from another realm and an anonymous user, which resulted in an authorization error. I also tried creating a flight assignment with a leg with no ID parameter, with ID=-1 and with an ID of a not published leg, for which I received authorization errors.

The modification of the moment was also tested, although not included in the testing file. When trying to change the moment and sending the form, it was not modified since the attribute is not bound. No issues or bugs were encountered during the recording of these testings.

Update

First, tests performing legal requests were performed. For that, I entered as member1 and accessed a flight assignment from one of the two listings. I first sent the form without modifications, and then tried every possible modification for each field with the facilitated sample data obtained from the ”Scrapbook/Sample-Data.xlsx” file. All turned into successful requests.

Secondly, tests performing illegal requests were performed. For it, I tried updating a flight assignment of member1 from another member, another realm and an anonymous user, which resulted in an authorization error. I also tried updating a flight assignment with a leg with no ID parameter, with ID=-1 and with an ID of a not published leg, for which I received authorization errors. To end with, I modified the URL in order to check the system response to an update request with no ID parameter, to ID=-1, to ID=, and to ID=XXX, for which I received authorization errors.

The modification of the moment was also tested, although not included in the testing file. When trying to change the moment and sending the form, it was not modified since the attribute is not bound. Apart from that, no issues or bugs were encountered during the recording of these testings.

Publish

First, tests performing legal requests were performed. For that, I entered as member1 and accessed a flight assignment from one of the two listings. I first sent the form without modifications, and then tried every possible modification for each field with the facilitated sample data obtained from the ”Scrapbook/Sample-Data.xlsx” file. All turned into successful requests.

However, member1 is assigned to overlapping legs, so that was tested as well with the realm, as well as publishing a flight assignment with a past leg. Later, I signed in as member2 and tried to publish a flight assignment, for which I received a validation error, since the member of the account is on leave. Then, I signed in as member3 and tried publishing a flight assignment that already had pilot and copilot roles, for which I received the corresponding validation errors. Finally, using two non overlapping legs, I created 2 assignments for member3, and published one of them. All this was done to check the validation explained in the requirement and obtain a 100 percent testing coverage.

Secondly, tests performing illegal requests were performed. For it, I tried publishing a flight assignment of member1 from another member, another realm and an anonymous user, which resulted in an authorization error. I also tried publishing a flight assignment with a leg with no ID parameter, with ID=-1 and with an ID of a not published leg, for which I received authorization errors. To end with, I modified the URL in order to check the system response to a publish request with no ID parameter, to ID=-1, to ID=, and to ID=XXX, for which I received authorization errors.

The validation had to be modified because it did not work correctly. Apart from that, no issues or bugs were encountered during the recording of these testings.

Delete

First, tests performing legal requests were performed. For that, I entered as member1 and accessed a flight assignment from one of the two listings. I deleted one and then accesed another one using the delete?id=*id* parameter in the URL.

Secondly, tests performing illegal requests were performed. For it, I tried deleting a flight assignment of member1 from another member, another realm and an anonymous user, which resulted in an authorization error. I also tried deleting a flight assignment with a leg with no ID parameter, with ID=-1 and with an ID of a not published leg, for which I received authorization errors. To end with, I modified the URL in order to check the system response to a delete request with no ID parameter, to ID=-1, to ID=, and to ID=XXX, for which I received authorization errors.

No issues or bugs were encountered during the recording of these testings.

3. Testing on activity logs

Activity logs had six features: list, show, create, update, delete and publish.

Listing

First, tests performing legal requests were performed. For that, I signed in as every flight crew member available and accessed the activity logs that were associated to flight assignments. I entered flight asignments activity logs' listings that actually contained some instances, as well as some empty listings.

Secondly, tests performing illegal requests were performed. For it, I tried accessing to the activity logs listings from another memeber, an anonymous user and a customer user, all giving authorization errors. I also modified the URL in order to check the system response to an list request with no masterId parameter, to masterId=-1, to masterId=, and to masterId=XXX, for which I received authorization errors.

No issues or bugs were encountered during the recording of these testings.

Show

First, tests performing legal requests were performed. For that, I entered into every activity log in all flight assignment listings for each flight crew member.

Secondly, tests performing illegal requests were performed. For it, I tried accessing an activity log of member1 from another flight crew member, another realm and an anonymous user, which resulted in an authorization error. I also modified the URL in order to check the system response to a show request with no ID parameter, to ID=-1, to ID=, and to ID=XXX, for which I received authorization errors.

No issues or bugs were encountered during the recording of these testings.

Create

First, tests performing legal requests were performed. For that, I entered as member1 and created a new activity log from the "Create" button in the listing. I first sent the form empty, and then I tried every possibility for each field with the facilitated sample data obtained from the "Scrapbook/Sample-Data.xlsx" file. Finally, I sent the form with valid data and successfully created the activity log.

Secondly, tests performing illegal requests were performed. For it, I tried creating an activity log from another realm and an anonymous user, which resulted in an authorization error. I also modified the URL in order to check the system response to an creation request with no masterId parameter, to masterId=-1, to masterId=, and to masterId=XXX, for which I received authorization errors.

No issues or bugs were encountered during the recording of these testings.

Update

First, tests performing legal requests were performed. For that, I entered as member1 and accessed a flight assignment from one of the two listings. Then I accessed an activity log from the former and sent the form without modifications. After that I tried every possible modification for each field with the facilitated sample data obtained from the "Scrapbook/Sample-Data.xlsx" file. All turned into successful requests.

Secondly, tests performing illegal requests were performed. For it, I tried updating an activity log of member1 from another flight crew member, another realm and an anonymous user, which resulted in an authorization error. I also modified the URL in order to check the system response to an update request with no ID parameter, to ID=-1, to ID=, and to ID=XXX, for which I received authorization errors.

No issues or bugs were encountered during the recording of these testings.

Publish

First, tests performing legal requests were performed. For that, I entered as member1 and accessed an activity log from one of the two listings. I first sent the form without modifications, and then tried every possible modification for each field with the facilitated sample data obtained from the "Scrapbook/Sample-Data.xlsx" file. All turned into successful requests. However, it could not be published, since the parent flight assignment was not published. Once done, the activity log was published succesfully.

Secondly, tests performing illegal requests were performed. For it, I tried publishing an activity log of member1 from another flight crew member, another realm and an anonymous user, which resulted in an authorization error. I also modified the URL in order to check the system response to an publish request with no ID parameter, to ID=-1, to ID=, and to ID=XXX, for which I received authorization errors.

There were some issues with the renderization of the "Publish" button when the request was "publish", and was quickly fixed in the unbind() methods of the update, delete and publish services. Apart from that, no issues or bugs were encountered during the recording of these testings.

Delete

First, tests performing legal requests were performed. For that, I entered as member1 and accessed an activity log from a flight assignment. I deleted one and then accesed another one using the delete?id=*id* parameter in the URL.

Secondly, tests performing illegal requests were performed. For it, I tried deleting an activity log of member1 from another member, another realm and an anonymous user, which resulted in an authorization error. I also modified the URL in order to check the system response to a delete request with no ID parameter, to ID=-1, to ID=, and to ID=XXX, for which I received authorization errors.

No issues or bugs were encountered during the recording of these testings.

4. Coverage and performance

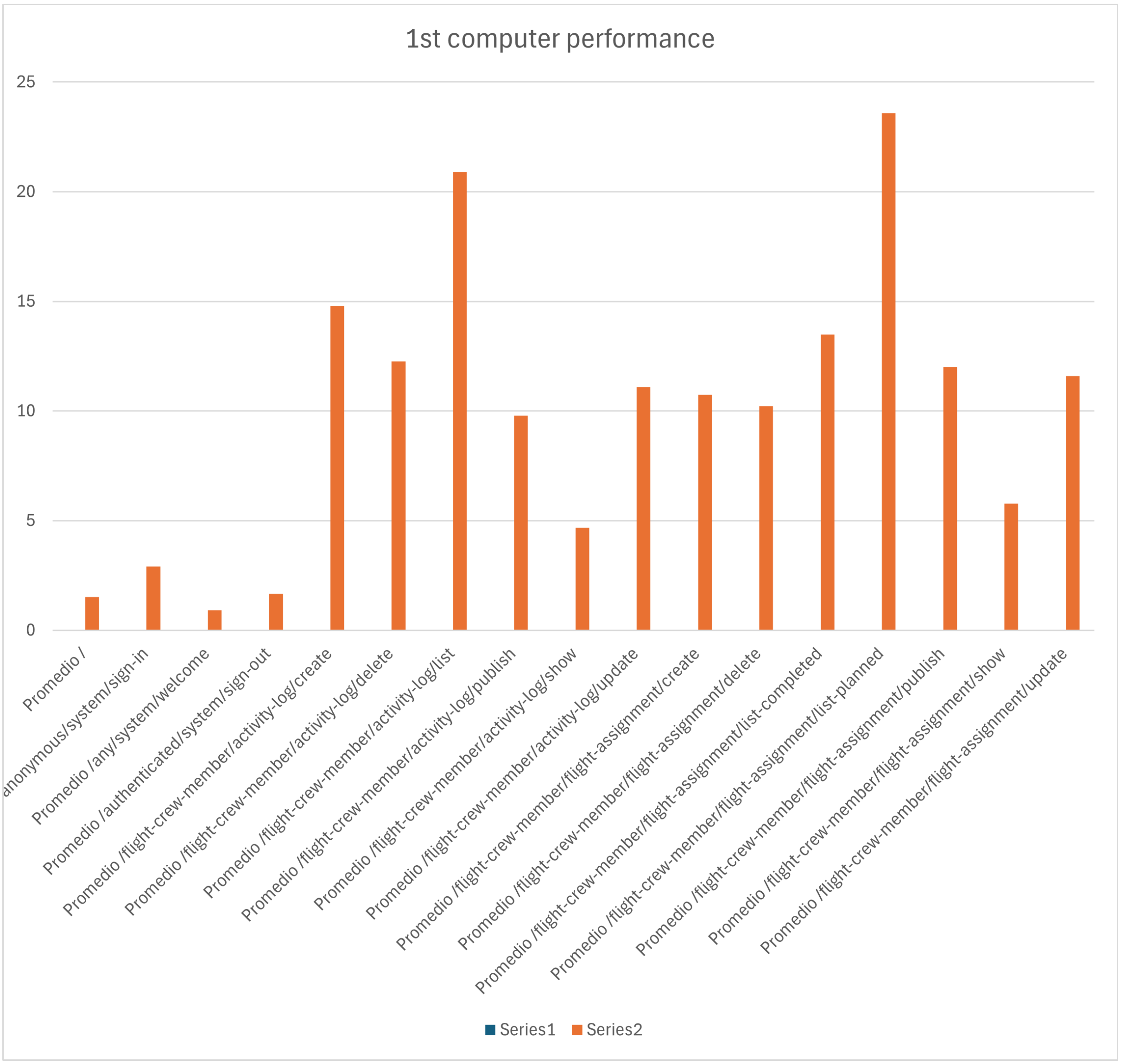
The whole testing suit regarding flight assignments and activity logs takes approximately 6 minutes and 21 seconds to execute completely. Coverage for both requirements is of 100 percent. There is only one conditional in the auth() methods of activity logs features which misses 1 out of 6 branches, since one of the included conditions (activity-Log.getAssignment() != null) will never be false.

As of performance, the test suit was executed in two different computers:

- MSI Katana with an i7 CPU, 16GB of RAM and a 500GB SSD.

- i7 CPU with 32GB of TAM and 1TB SSD.

This is the graph obtained for the first computer:



As it can be seen, the MIR (most inneficient request) is /flight-crew-member/flight-assignment/list-planned, taking 23,58637021 miliseconds.

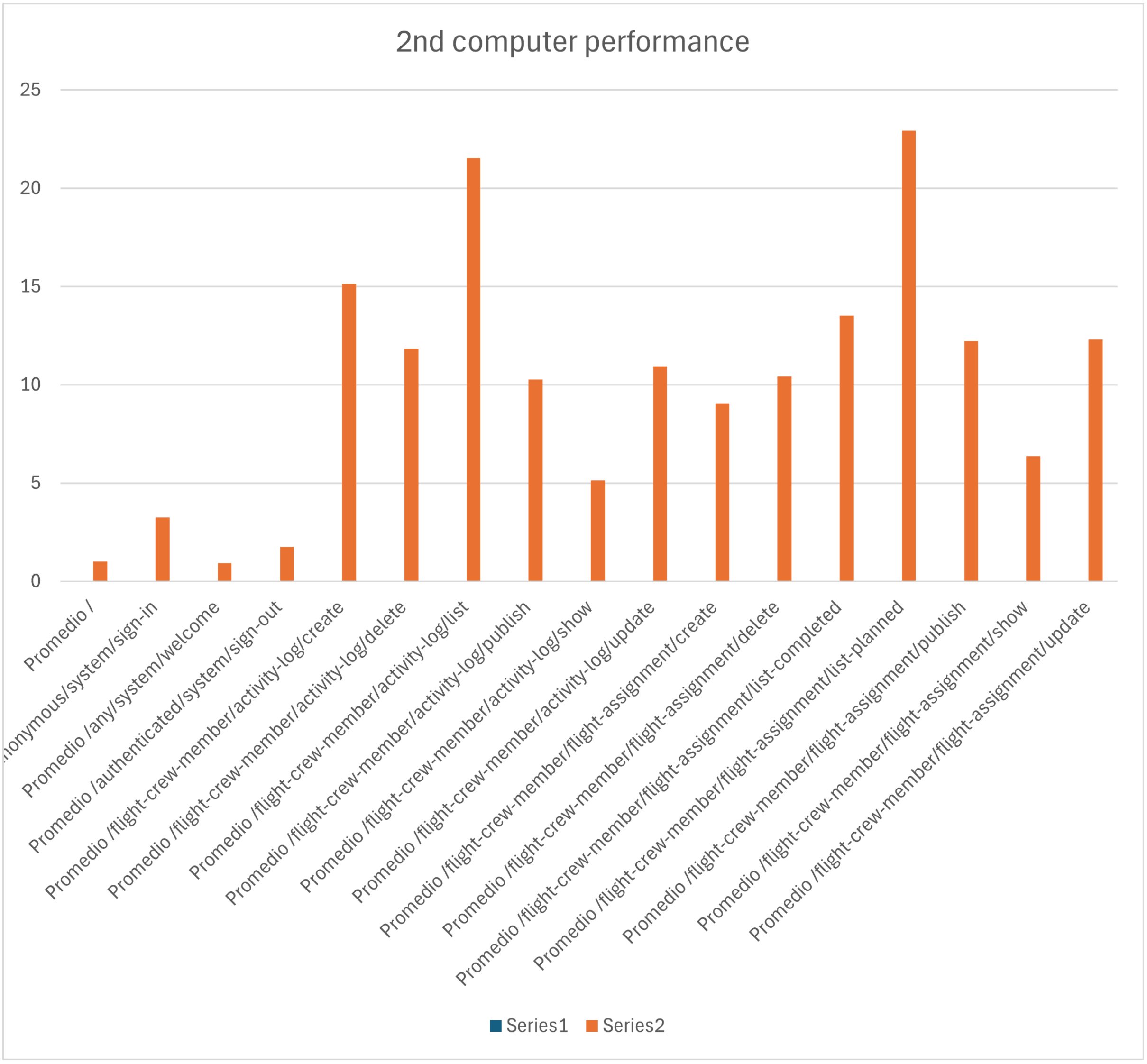
After performing a statistical analysis, I obtained the following conclusions:

- The average time for the first computer is 6,985652913 ms.

- The 95 percent confidence level is 0,450040952.

- The 95 percent confidence interval is [6,535611962, 7,435693865]

This is the graph obtained for the second computer:



As it can be seen, the MIR (most inneficient request) is /flight-crew-member/flight-assignment/list-planned, taking 23,58637021 miliseconds. The request /flight-crew-member/activity-log/list almost takes the same time (21,53369896 ms).

After performing a statistical analysis, I obtained the following conclusions:

- The average time for the second computer is 7,307162776 ms.

- The 95 percent confidence level is 0,466306663.

- The 95 percent confidence interval is [6,840856112, 7,773469439]

After performing a Z-test, I obtained a p-value of 0,288033775017444, which is bigger than 0.05, meaning that none of them have significant improvements compared to one another (they are globally the same).

5. Conclusion

In conclusion, the test suite provided for this delivery appears to be complete. It covers not only legal requests but also illegal requests and potential hacking attempts, ensuring robustness.

The testing performed has significantly contributed to validating the correctness, security, and reliability of the implemented features. It has also facilitated the detection of unnintended bugs that were corrected later. Thanks to the documentation of this test suite, we can provide the customer with the same level of confidence that we have in the quality and reliability of the work delivered.

References

Intentionally blank