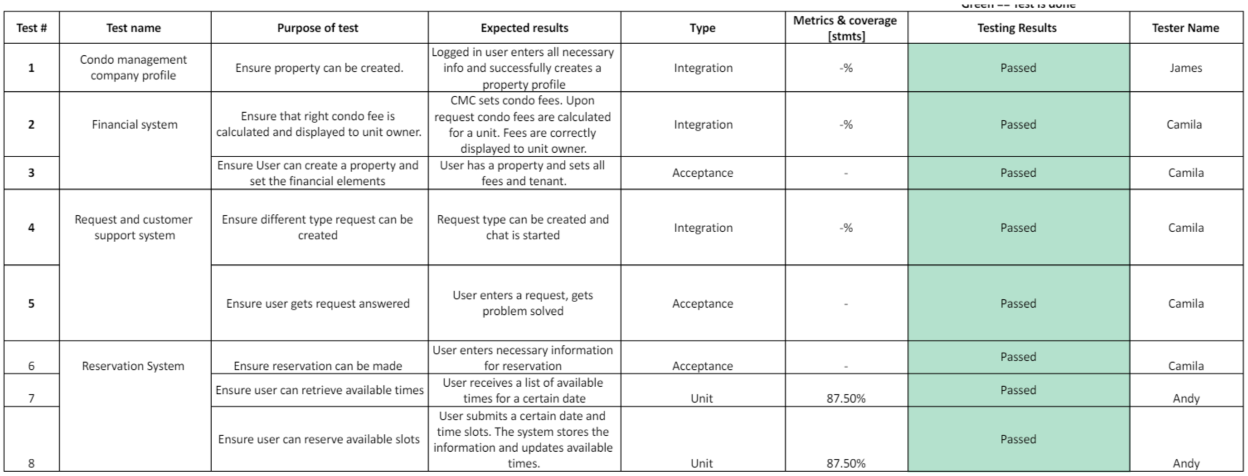
Testing Report

Overview

In this sprint, we wrote 8 tests. We did unit testing and continued with end-to-end testing. We were able to enable code coverage with Cypress but there are still fixes to make. The coverage does not currently cover the right files but we will still include a sample of what it looks like. We will improve this code coverage feature for the next sprint.



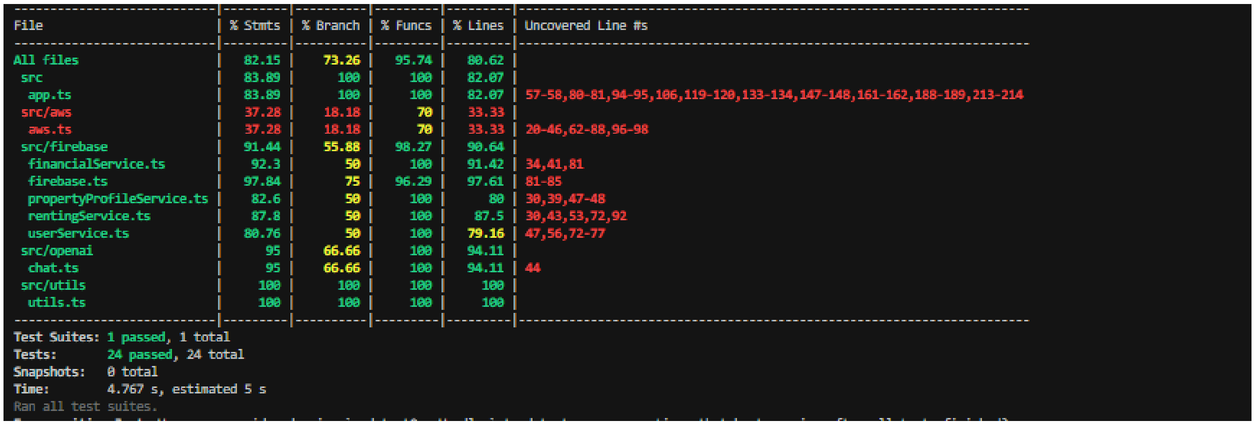
Green == Test is done

Test #	Test name	Purpose of test	Expected results	Type	Metrics & coverage [stmts]	Testing Results	Tester Name
1	Condo management company profile	Ensure property can be created.	Logged in user enters all necessary info and successfully creates a property profile	Integration	-%	Passed	James
2	Financial system	Ensure that right condo fee is calculated and displayed to unit owner.	CMC sets condo fees. Upon request condo fees are calculated for a unit. Fees are correctly displayed to unit owner.	Integration	-%	Passed	Camila
3		Ensure User can create a property and set the financial elements	User has a property and sets all fees and tenant.	Acceptance	-	Passed	Camila
4	Request and customer support system	Ensure different type request can be created	Request type can be created and chat is started	Integration	-%	Passed	Camila
5		Ensure user gets request answered	User enters a request, gets problem solved	Acceptance	-	Passed	Camila
6	Reservation System	Ensure reservation can be made	User enters necessary information for reservation	Acceptance	-	Passed	Camila
7		Ensure user can retrieve available times	User receives a list of available times for a certain date	Unit	87.50%	Passed	Andy
8		Ensure user can reserve available slots	User submits a certain date and time slots. The system stores the information and updates available times.	Unit	87.50%	Passed	Andy

Unit Testing

As shown above, the unit testing was focused on the reservation system story.

From the snapshot included below, we can see that the tests achieved 80%+ coverage.



```
----------------------------|---------|----------|---------|---------|----------------------------------------------------------------------
File                        | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
----------------------------|---------|----------|---------|---------|----------------------------------------------------------------------
All files                   |   82.15 |    73.26 |   95.74 |   80.62 |
 src                        |   83.89 |      100 |     100 |   82.07 |
  app.ts                    |   83.89 |      100 |     100 |   82.07 | 57-58,80-81,94-95,106,119-120,133-134,147-148,161-162,188-189,213-214
 src/aws                    |   37.28 |    18.18 |      70 |   33.33 |
  aws.ts                    |   37.28 |    18.18 |      70 |   33.33 | 20-46,62-88,96-98
 src/firebase               |   91.44 |    55.88 |   98.27 |   90.64 |
  financialService.ts       |    92.3 |       50 |     100 |   91.42 | 34,41,81
  firebase.ts               |   97.84 |       75 |   96.29 |   97.61 | 81-85
  propertyProfileService.ts |    82.6 |       50 |     100 |      80 | 30,39,47-48
  rentingService.ts         |    87.8 |       50 |     100 |    87.5 | 30,43,53,72,92
  userService.ts            |   80.76 |       50 |     100 |   79.16 | 47,56,72-77
 src/openai                 |      95 |    66.66 |     100 |   94.11 |
  chat.ts                   |      95 |    66.66 |     100 |   94.11 | 44
 src/utils                  |     100 |      100 |     100 |     100 |
  utils.ts                  |     100 |      100 |     100 |     100 |
----------------------------|---------|----------|---------|---------|----------------------------------------------------------------------
Test Suites: 1 passed, 1 total
Tests:       24 passed, 24 total
Snapshots:   0 total
Time:        4.767 s, estimated 5 s
Ran all test suites.
```

* The 37% coverage seen is from a database issue, it does not represent the tests and so we still consider all of our tests to have successfully passed.

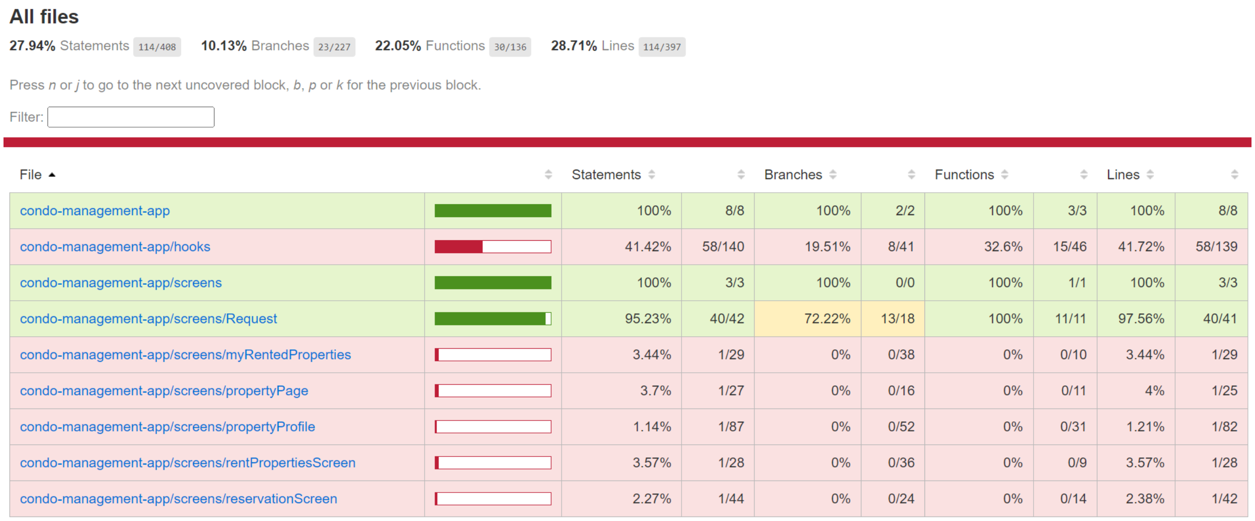
Integration Testing

The following integration tests were completed:

- Adding a property (https://youtu.be/vKL7TgvR8Ww)
- Editing existing property (https://youtu.be/o_r3t_to1Zw)
- Creating a Request (https://youtu.be/ZoNRqN6HVEE)

Coverage

Here we are including a sample of the coverage generated by Cypress. It still needs to be fixed, but it is at least working for this sprint.



All files

27.94% Statements 114/408 **10.13%** Branches 23/227 **22.05%** Functions 30/136 **28.71%** Lines 114/397

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

Filter:

File ▲		Statements		Branches		Functions		Lines	
condo-management-app		100%	8/8	100%	2/2	100%	3/3	100%	8/8
condo-management-app/hooks		41.42%	58/140	19.51%	8/41	32.6%	15/46	41.72%	58/139
condo-management-app/screens		100%	3/3	100%	0/0	100%	1/1	100%	3/3
condo-management-app/screens/Request		95.23%	40/42	72.22%	13/18	100%	11/11	97.56%	40/41
condo-management-app/screens/myRentedProperties		3.44%	1/29	0%	0/38	0%	0/10	3.44%	1/29
condo-management-app/screens/propertyPage		3.7%	1/27	0%	0/16	0%	0/11	4%	1/25
condo-management-app/screens/propertyProfile		1.14%	1/87	0%	0/52	0%	0/31	1.21%	1/82
condo-management-app/screens/rentPropertiesScreen		3.57%	1/28	0%	0/36	0%	0/9	3.57%	1/28
condo-management-app/screens/reservationScreen		2.27%	1/44	0%	0/24	0%	0/14	2.38%	1/42

As seen in the image above, the coverage numbers are very low, but with the testing done, this does not seem to be representative of what we have done. We plan to fix and improve this for the last sprint.

Acceptance Testing

UAT1- Creating Property and setting fees

Description: User can create a property profile and set the corresponding fees

Prerequisites: User must have an account and be signed in.

Test Steps:

1- User clicks on “Property profiles” tab

2- User click on “+” to add property

3- User enters the property information on the text fields.

4- User clicks on “Add Property”

Expected result:
Property created is added at the bottom of the page.

Actual result:
Same as expected

Status:
Pass [X]
Fail [ ]

UAT2- Make a request
Description: User can make a request for maintenance or a complaint.
Prerequisites: User must have an account and be signed in.
Test Steps:
1- User clicks on “Make a Request” tab.
2- User selects request type from dropdown.
3- User enters the issue in the Message textbox
4- User clicks on “Send”
5- User provides more information to chat employee.
6- User clicks on “End Chat” when issue is resolved.

Expected result:
Chat is opened upon request creation and issue is resolved by the employee.

Actual result:
Same as expected

Status:
Pass [X]
Fail [ ]

UAT3- Make a reservation
Description: User can make a reservation on a rented property
Prerequisites: User must have an account and be signed in.
Test Steps:
1- User clicks on “Rent a Property” tab
2- User chooses property and clicks on “Start Renting” button
3- User clicks on “My Rented Properties” tab
4- User clicks on “Make a Reservation” button for the desired service.
5- User selects date for reservation and clicks on “save” button
6- User selects timeslot for reservation.
7- User clicks “Reserve” button

Expected result:
Reserved activity is displayed under “My displayed activities.

Actual result:
Same as Expected.

Status:
Pass [X]
Fail [ ]

Improvements

For the next sprint, we plan to fix and improve the code coverage on Cypress, as well as finalize the testing for our main features. We were also planning to have an integration test done to show the reservation system, but we found that Cypress does not recognize the calendar widget we used. This made it so that we could not finish that test even though we knew it passed the acceptance test. Our plan for the next sprint is to fix this or change this widget to correctly test these features.