Testing Report

Overview

In this sprint, we wrote 11 tests. We mostly focused on finishing up our features.

Test #	Test name	Purpose of test	Expected results	Type	Metrics & coverage [stmts]	Testing Results	Tester Name
1	Financial system	Ensure CMCs can record operational budgets and then generate a financial report	CMC updates operational budgets. CMC requests for annual report. Annual report is generated.	Integration	80%	Passed	Camila
2		Ensure CMCs can record operational budgets and then generate a financial report	CMC updates operational budgets. CMC requests for annual report. Annual report is generated.	Acceptance	- %	Passed	Camila
3	Reservation System	Ensure reservation can be made	User enters necessary information for reservation	Integration	-%	Cypress bug, manual testing works	Camila
4		Ensure User can make a reservation	User enters information and makes a reservation successfully	Acceptance	-%	Passed	Camila
5	Employee Role Management system	Ensure employee type can be chosen and saved	User choses a valid employee type	Unit	81%	Passed	Andy
6		Ensure employer is authenticated before removing an employee	User is rejected if removing an employee is done unauthenticated	Unit	87%	Passed	Andy
7		Ensure employer is authenticated before adding an employee	User is rejected if addingan employee is done unauthenticated	Unit	87%	Passed	Andy
8		Employer is able to get the list of potential employees	Employer is authenticated and gets the list of potential employees	Unit	87%	Passed	Andy
9		Employer is able to get the list of currently employed users	Employer is authenticated and gets the list of current employees	Unit	87%	Passed	Andy
10		Employer is able to add an employee	Employer is authenticated and adds to the list of current employees	Unit	87%	Passed	Andy
11		Employer is able to remove an employee	Employer is authenticated and deletes from the list of current employees	Unit	87%	Passed	Andy

Unit Testing

As shown above, the unit testing was focused on the employee role story.

From the snapshot included below, we can see that the tests achieved 81%+ coverage.

```
File                        | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
----------------------------|---------|----------|---------|---------|-------------------
All files                   |  83.01  |   73.52  |  93.98  |  81.64  |
 src                        |  81.92  |   96.72  |  94.73  |  80.37  |
  app.ts                    |  81.92  |   96.72  |  94.73  |  80.37  | 69-70,92-93,106-107,118,131-132,140-141,154,162,175-176,189-190,203-204,230-231,255-256,269-270,283-284,297-298,311-312
 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               |  91.08  |   59.25  |  96.38  |  90.24  |
  employeeManagement.ts     |  87.87  |      50  |    100  |  87.09  | 17,27,41,55
  financialService.ts       |   92.3  |      50  |    100  |  91.42  | 34,41,81
  firebase.ts               |  95.33  |   70.58  |  93.18  |  94.81  | 84-88,145-153
  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            |  82.14  |   66.66  |    100  |  80.76  | 51,65,81-86
 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:       37 passed, 37 total
Snapshots:   0 total
Time:        5.771 s, estimated 7 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:

- Add Cost Entry and Generate Report (https://youtu.be/OE76DAoZ1aU)

We also made an integration test to test the reservation process but we encountered a bug between Cypress and the calendar component we used to set the date (datepicker modal). It was tested manually and we are also including an acceptance test for it.

Acceptance Testing

UAT1- Add cost entry and generate report
Description: User is logged in and enters a cost for a specific tenant. User can then generate a pdf report of these costs.
Test Steps:
1- User clicks on "Property Profiles" tab button
2- User clicks on the "Financials" button of a specific property
3- User enters the tenant ID, the cost description and cost into the text inputs.
4- User clicks on "Add Cost Entry" button
5- User clicks on "Generate report" button

Expected result:
Cost entry is added and pdf report is generated with the costs added.

Actual result:
Same as expected

Status:
Pass [X]
Fail   [   ]

UAT2- Make a reservation
Description: User is logged in and can reserve an activity in a rented property.
Test Steps:
1- User clicks on "My Rented Properties" tab button
2- User clicks on "Reserve" button for the activity wanted in the specific property.
3- User selects date in calendar and clicks on the "save" button
4- User selects the time period for the reservation
5- User clicks on "Reserve" button

Expected result:
Reservation info appears in "My reserved activities" below

Actual result:

Same as expected

Status:
Pass [X]
Fail   [   ]

Improvements

For future sprints, we would work on the cypress coverage and maybe consider other tools. We would also consider doing system tests now that we have a functional system.