SPARTA TEST PLAN FOR CODEBAR

SDET13

KA LOK CHEUNG | SORAIA CARMO | NARULLAH NOOR

Table of Contents

1. Introduction

This test plan has been created to communicate the test approach to team members. It includes the objectives, scope, schedule, risks and approach. This document will identify what the test deliverables will be and what is deemed in and out of scope.

2. Objectives

Create an automation suite for Codebar. The automation suite will test the user interface and the flow of the website to test for any errors.

Team Members:

Resource Name
Ka Lok Cheung
Soraia Carmo
Narullah Noor

3. Scope

The goal is to find what other features are left to test and make it as a Page Object Model (POM). Previous tests that have been done, may be added to the POM as well.

This will help improve the readability of the code as well as re using methods that have already been defined in the pages.

Main features of the page to be tested:

- Signing In
- Signing Up
- Signing Out
- Ability to attend workshops

Functional Testing

- Testing web-app on Chrome
- Use case creation
- Regression Testing

Non-Functional Testing

- Usability Testing
- Exploratory Testing

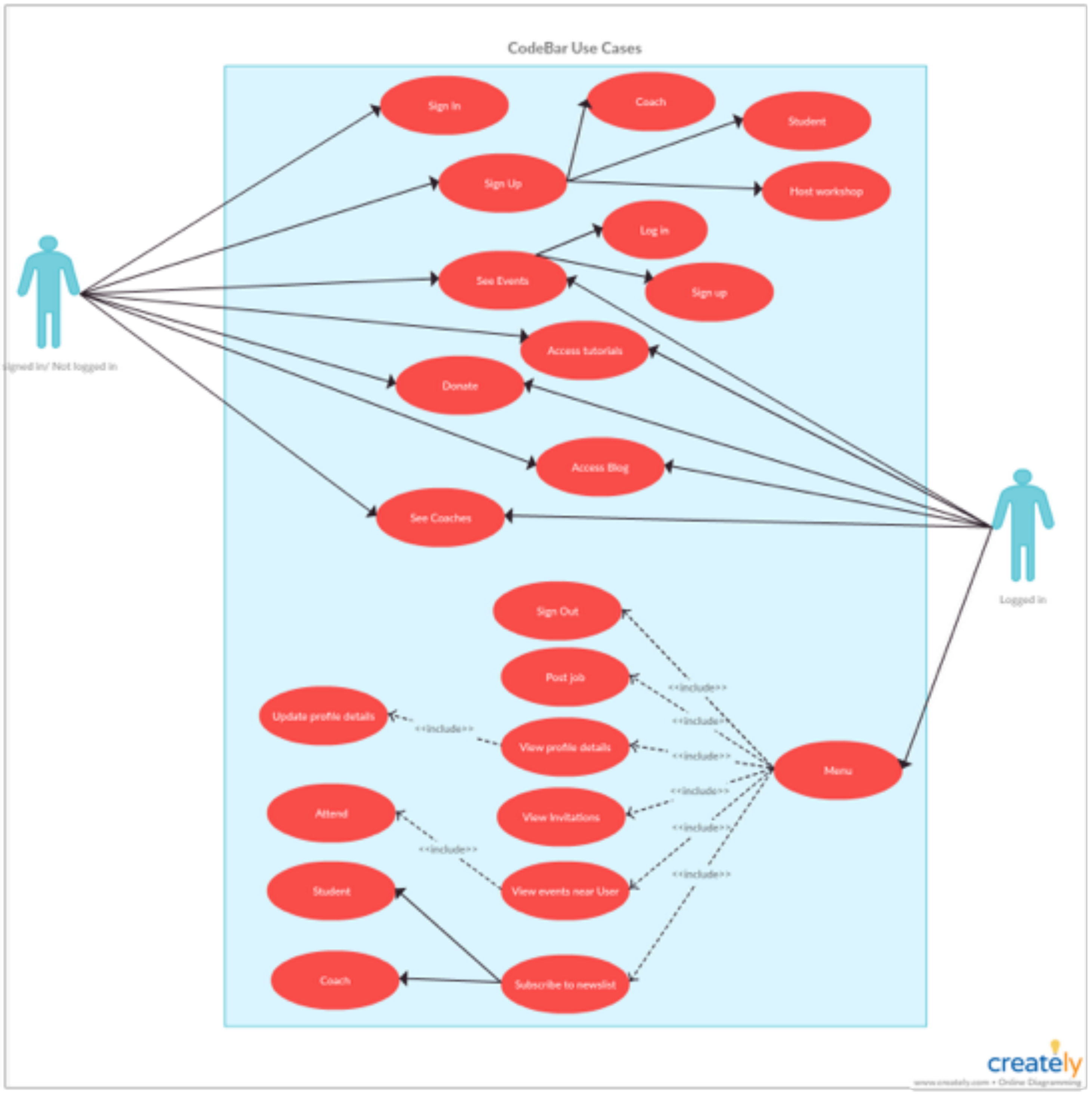


Figure. 1 Use Case Diagram for Codebar

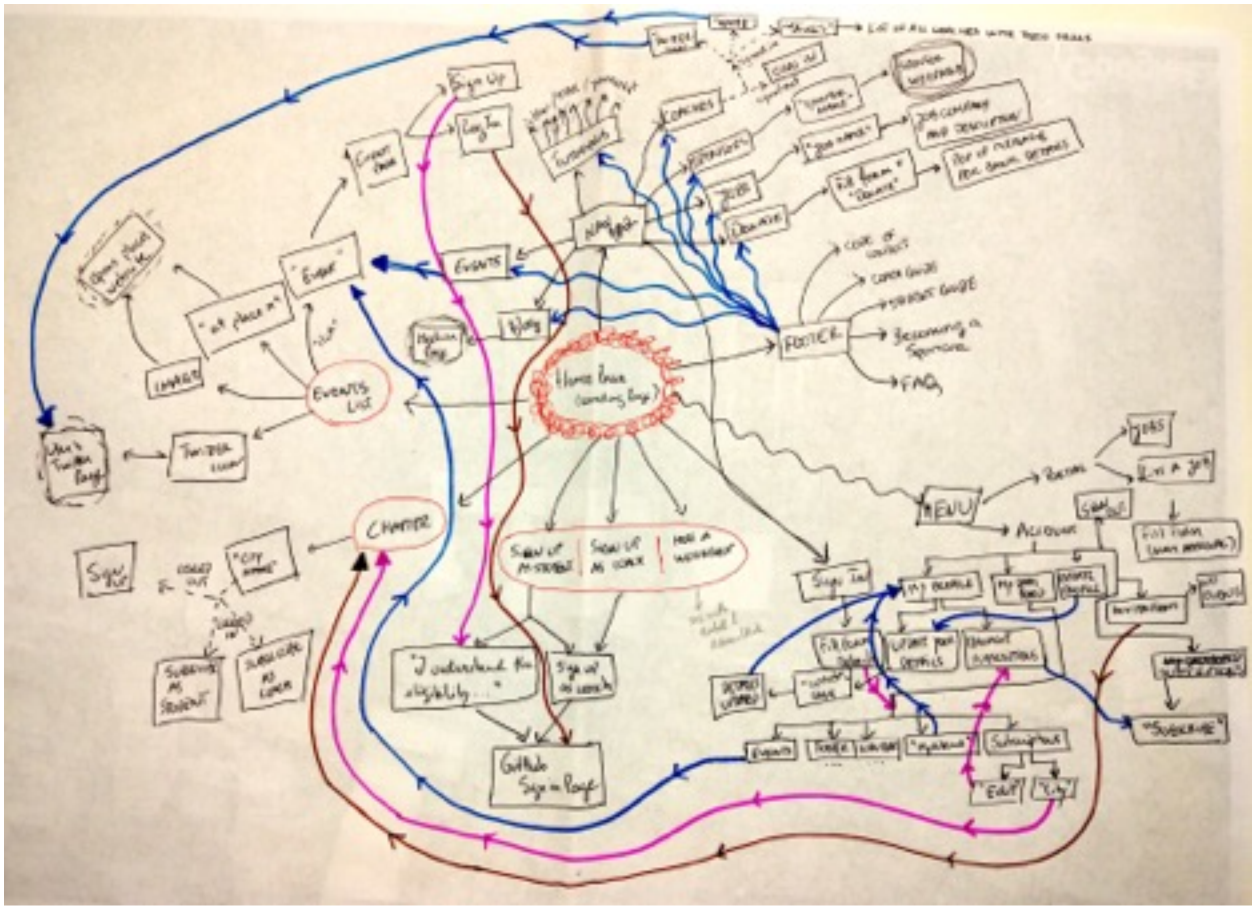


Figure. 2 A basic sitemap of Codebar

4. Assumptions

4.1 Assumptions

No business requirements have been given, so the team had to study and assume some of the requirements and features on the website. The features have been described in the Scope.

4.2 Risks

When testing on actual website, an actual booking will be made to attend the workshop.

5. Test Approach

The project was completed using an agile approach, with weekly sprints. At the end of each week the requirements identified for that sprint will be delivered by the team and will be tested.

Exploratory testing will play a large part of the testing as the team has never been or used the website and will be learning as they go.

Go to https://github.com/Kalok9990/planner to look at the code.

5.1 Test Automation

Capybara will be used to perform the majority of the tests.

6. Test Environment

A new database needs to be created to test locally. All instructions are given in the ReadMe document to set up the environment. The tests all run on Google Chrome on MAC OSX. Test data is generated and can be seen in the ReadMe file.

7. Tools

- MacBook Air with Google Chrome
- Trello
- Google Drive
- Pages to write test plan
- Atom to write tests with
- Cucumber

8. Deliverables

8.1 Test Schedules

The initial test schedule follows:

Task Name	**Date Due**
Test Planning	16/01/2018
Create initial test estimates	16/01/2018
Create Test Cases	18/01/2018
UAT	23/01/2018
Report	24/01/2018

9. Entry and Exit Criteria

Entry

- Test plan signed off
- User stories created
- Test cases produced

Exit

- All test cases completed
- Defects documented in the final report

10. Approvals

Full Name	Role in Organisation	Date	Date Approved
Richard Gurney	Project Manager	16/01/2018	16/01/2018

11. Report

11.1. Test Reports

The majority of the user journeys on the website goes to the correct pages, however the content in the coaches and sponsors page have not been tested yet.

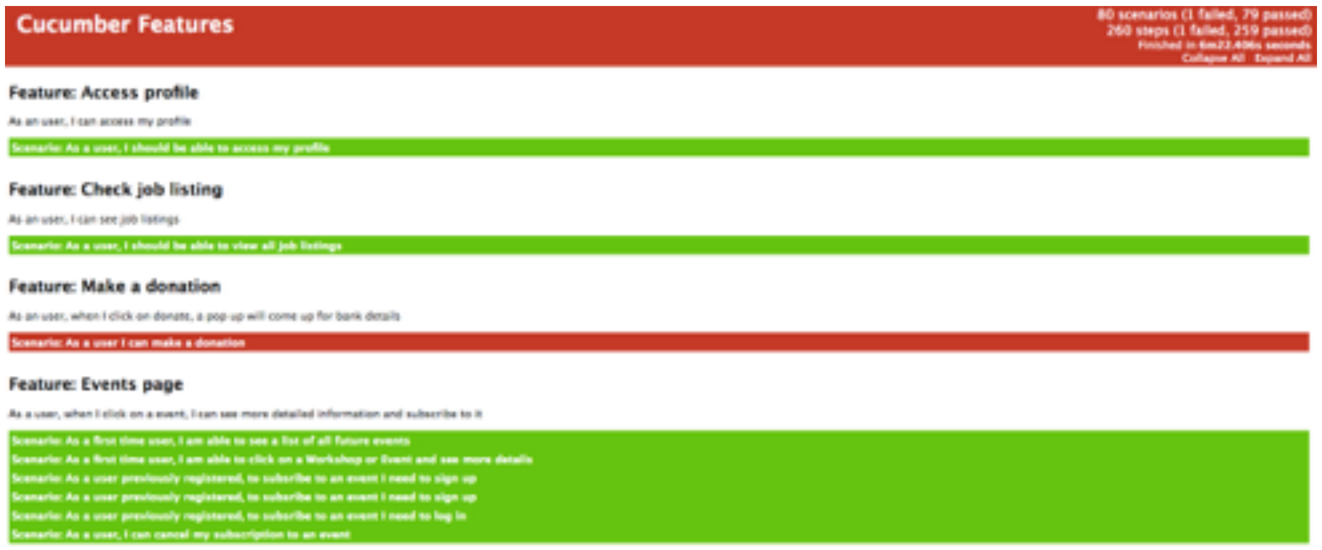
11.1.1. Difficulties

Some difficulties arose when some tests were implemented. This was due to the flow of the website where it authorises against Github when a user tries to log in.
Trying to test the logic of the website proved to be very difficult as different pages would be retrieved during automation testing compared to when the website was manually explored.

Signing in to the website proved to be interesting as well. Once you have signed in and signed out, when you try to sign back in again, it will automatically resign you into the account that you had previously signed in with. This took a while to realise that it was due to Codebar authorising against Github. If you have a Github account signed in, then Codebar will automatically signup to that account.

11.1.2. Cucumber Report

To have a look at the tests that have been run, clone the repository and go to
> planner/ui-testing/features/reports then click on report.html which will open up a clean format of the tests that have been run.
Please refer to Figure 3 for a screenshot of the cucumber report.



Cucumber Features

80 scenarios (1 failed, 79 passed)
260 steps (1 failed, 259 passed)
Finished in 6m22.406s seconds
Collapse All Expand All

Feature: Access profile

As an user, I can access my profile

Scenario: As a user, I should be able to access my profile

Feature: Check job listing

As an user, I can see job listings

Scenario: As a user, I should be able to view all job listings

Feature: Make a donation

As an user, when I click on donate, a pop up will come up for bank details

Scenario: As a user I can make a donation

Feature: Events page

As a user, when I click on a event, I can see more detailed information and subscribe to it

Scenario: As a first time user, I am able to see a list of all future events
Scenario: As a first time user, I am able to click on a Workshop or Event and see more details
Scenario: As a user previously registered, to subscribe to an event I need to sign up
Scenario: As a user previously registered, to subscribe to an event I need to sign up
Scenario: As a user previously registered, to subscribe to an event I need to log in
Scenario: As a user, I can cancel my subscription to an event

Figure 3 - Cucumber’s Report Screenshot

11.2. Bug Reports

Defect ID	**Defect Title/Description**	**Steps to Reproduce**	**Severity (1-10)**
1	Can sign in to different account, however it is a very long winded progress of having to sign out of your GitHub account on GitHub before signing in to a different account. Not very user friendly	"1) Sign in on one account 2) Sign out of the account 3) Click Sign in"	6
2	When pressing the donate button, does not always bring up the pop up	"1) Go to the Donate page 2) Enter name and amount 3) Click Donate 4) Repeat until failure"	3

11.3. Improvements

The website itself is fine in where all user journeys tested go to the correct pages and displays the appropriate information. However as mentioned previously in 11.1.1., the flow of the website is not great. This is mainly due to issue of logging in/ signing in.
If the log in process could be improved by going to the log in form each time, rather than automatically signing in to the used Github account, then that may make it smoother.

12. Summary

During this project all the explored and proposed user journeys (Figure. 1) were tested.
The team attempted to test majority of the sitemap as seen in Figure 2, however as exhaustive testing is impossible, it was not able to test the entire website.
There were several difficulties at the start and for that exploratory testing was carried out.

Some issues that were found: when trying to donate, the form to enter the bank details does not always pop up or takes a lot of clicks before it pops up.
When trying to sign in, it is not very user friendly as it will always sign up to the Github account that was previously logged in to the computer.
Overall the user interface is running fine and functions as it should do, but with just a few issues that will be reported to Codebar development team.